PC-8011B
Expansion Unit
Reference Manual

NEC

PC-8106B
PTS-071

PC-8011B Expansion Unit Reference Manual

NEC

IMPORTANT NOTICE

(1) All rights reserved. This manual is protected by copyright. No part of this manual may be reproduced in any form whatsoever without the written permission of the copyright owner.

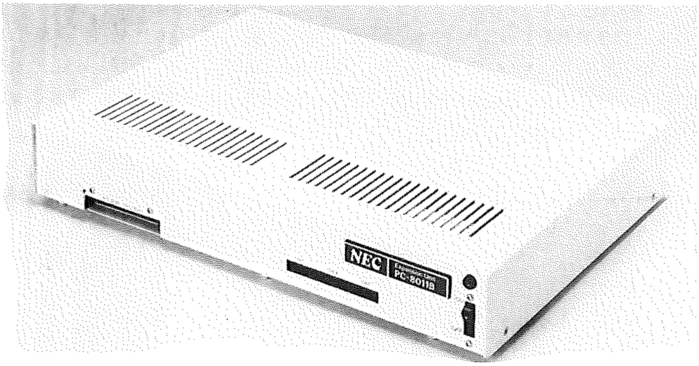
(2) The policy of NEC being that of continuous product improvement, the contents of this manual are subject to change, from time to time, without notice.

(3) All efforts have been made to ensure that the contents of this manual are correct; however, should any errors be detected, NEC would greatly appreciate being informed.

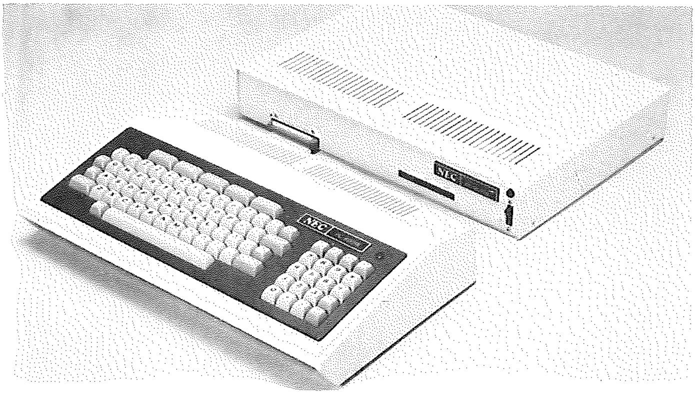
(4) NEC can assume no responsibility for errors in this manual or their consequences.

PREFACE

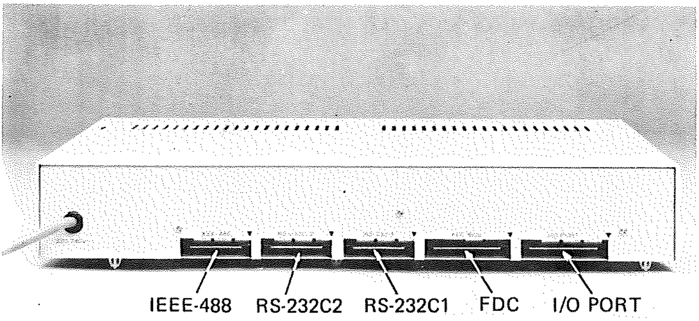
This manual covers the basic principles of operation and hardware design of the NEC PC-8011 Expansion Unit. It should be read by anyone connecting the PC-8011 to a PC-8001 Computer.



PC-8011 Expansion Unit



PC-8001 Computer with PC-8011 Connected



PC-8011 Back View

CONTENTS

Page

CONTENTS (continued)

Page

CONTENTS (continued)

Page

CONTENTS (continued)

Page

APPENDIX C μPB8214-C PRIORITY INTERRUPT CONTROLLER (continued)

CONTENTS (continued)

FIGURES

Page

FIGURES (continued)

TABLES

CHAPTER 1

FEATURES OF THE PC-8011 EXPANSION UNIT

The features of the PC-8011 are described below.

32K RANDOM ACCESS MEMORY

The PC-8011 contains 32K of Random Access Memory (RAM) of the NEC μPD416C-3 type and provides for the expansion of the PC-8001 memory map to a full 64K RAM.

8K READ ONLY MEMORY

The PC-8011 contains four sockets that accept NEC μPD2716 Read Only Memories/Programmable Read Only Memories (ROMs) or (PROMs).

INTERRUPT CAPABILITY

The PC-8011 has provisions for 16 levels of priority interrupts. Ten of these levels are available to you.

RS-232-C INTERFACE

Two channels of serial I/O are available. These channels are interrupt-driven NEC μPD8251C type Universal Synchronous/Asynchronous Receiver/Transmitters (USART) that operate at a baud rate of 4800, 2400, 1200, 600, 300, 150, or 75. The PC-8011 provides a 127-character buffer.

MINI-DISK CONTROLLER PORT

The expansion unit provides a controller port for the NEC PC-8031 Diskette Drive unit. This controller port consists of one NEC μPD8255AC-5 input/output (I/O) integrated circuit (IC).

PARALLEL I/O PORTS

Four parallel I/O ports are available. These ports are all Transistor to Transistor Logic (TTL) compatible. They are arranged as follows.

- One 8-bit input port
- One 8-bit output port
- One 4-bit input port
- One 4-bit output port

The fan-out on the output is ten LSTTL loads.

IEEE-488 (GP-IB) INTERFACE

The PC-8011 has an IEEE-488 interface using an NEC μPD8255AC-5 IC and TTL logic. The interface uses the following subset: SH1, AH1, T6, L4, SR1, RLO, PPO, DCO, DTO, C1, C2, C3, C4, and C28. The interrupt capability is on ATN, SRQ.

REAL-TIME CLOCK

The PC-8011 has a real-time clock. The clock period is 1.67 ms (600 Hz). It generates a maskable interrupt.

I/O BUS

The I/O provided by the PC-8011 allows I/O expansion using addresses 80H through 0AFH. This bus is TTL-compatible and has a fan-out of ten TTL loads.

POWER SUPPLY

The PC-8011 has an internal power supply that operates on 220/240 V ∼±10%, 50/60 Hz. Power consumption is 35 watts.

Precautions with the Power Supply

It is important that you take the following precautions.

- After turning off the power supply, wait at least five seconds before turning it back on. In the same manner, if the PC-8011 is unplugged with the power switch on, wait at least five seconds before plugging it back in.
- Make sure that the power supplied to the PC-8011 is 220/240 V ~ at 50/60 Hz.
- Always grip the plug when plugging and unplugging the expansion unit.

ENVIRONMENT REQUIREMENTS

The PC-8011 will operate over a temperature range of 32° to 95°F (0° to 35°C) and humidity range of 20% to 80% without condensation. The nonoperating storage temperature range is 5° to 140°F (-15° to 60°C).

Precautions Regarding Environmental Conditions

It is important that you take the following precautions.

- To prevent heat build-up, the case of the PC-8011 has air vents. Do not block these vents or operate the PC-8011 in a place where the air circulation is poor. Do not store the PC-8011 in extremely hot or cold places or in places subject to extreme temperature changes.
- Do not store or operate the PC-8011 in direct sunlight or near sources of heat.
- Do not store or operate the PC-8011 in especially dusty or moist places.
- Do not subject the PC-8011 to strong shocks or vibrations.

- Do not operate the PC-8011 with any foreign materials, especially liquids or metallic objects, inside the case.
- Do not store or operate the PC-8011 where it will come in contact with any strong chemicals or their vapors.

BLOCK DIAGRAM

An overall block diagram is given in Figure 1-1.

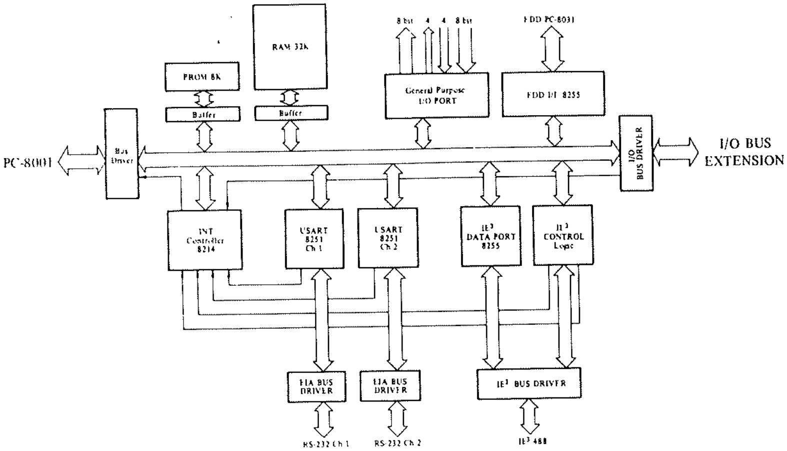


Figure 1-1 PC-8011 Block Diagram

DIMENSIONS

The dimensions for the PC-8011 follow.

Width:	16.92 inches (430 mm)
Depth:	11.61 inches (295 mm)
Height:	2.95 inches (75 mm)
Weight:	9.92 lbs. (4.5 kg)

CAUTIONS

Do not use strong chemicals to clean the case. Clean it only with a soft cloth moistened with water or a little detergent.

In case of malfunction or any obvious signs of trouble, such as overheating, contact the PC-8011 dealer.

CHAPTER 2
MEMORY

The PC-8011 Expansion Unit expands the memory map of the PC-8001 computer. One 8K ROM area and one 32K RAM area are available within the PC-8011. This memory is added to the PC-8011 in one of four available modes (see Figure 2-1.

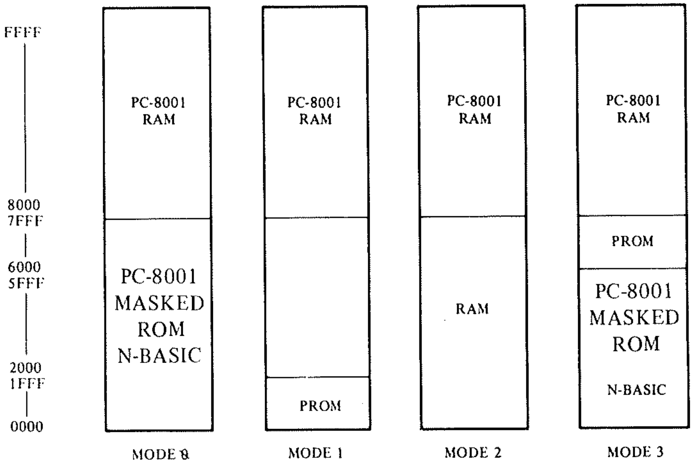


Figure 2-1 Memory Modes

MODE 0

In Mode 0, all of the memory within the PC-8001 is active. None of the memory in the PC-8011 is active. N-Basic will run in this mode.

MODE 1

In Mode 1, the first ROM in the PC-8001 is disabled. In its place, the ROM area within the PC-8011 is enabled. The ROM area of the PC-8011 occupies addresses 0000 to 1FFF in the PC-8001 memory map. N-Basic will not run in this mode.

MODE 2

In Mode 2, all of the ROM in the PC-8001 is disabled and replaced by the 32K RAM of the PC-8011. N-Basic will not run in this mode.

MODE 3

In Mode 3, the fourth ROM in the PC-8001, located at addresses 6000 to 7FFF, is disabled. In its place is the ROM area within the PC-8011. N-Basic will run in this mode.

NOTE

The computer can write the data and the program to the 32K RAM.

MODE SELECTION

You can select the operating mode of the PC-8011 by DIP switch or through software.

Mode Selection by DIP Switch

The mode of operation that the PC-8011 will enter when powered up or reset is determined by the setting of DIP switch "SW1" (see Figure 2-2). The proper switch settings for each mode are shown in Table 2-1.

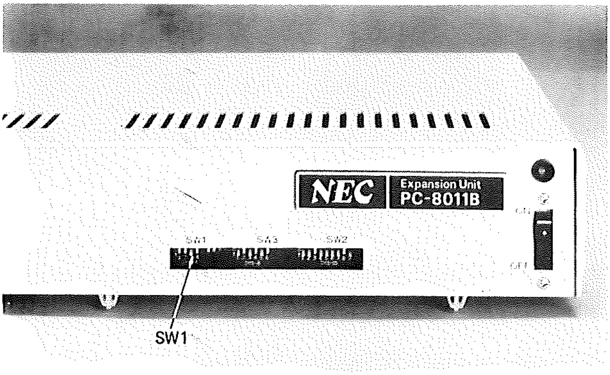


Figure 2-2 Switch SW1 Location

Because the restart address for all modes is 0, a power-up or reset causes the PC-8001 to enter N-Basic in Modes 0 and 3. In Mode 1, the PC-8001 enters the program contained in the ROM area of the PC-8011. Mode 2 cannot be chosen for auto-start after power-up or reset.

Table 2-1 SW1 Settings

2	1	MODE
ON	ON	MODE 0
ON	OFF	MODE 1
OFF	ON	UNDEFINED ON POWER-UP (Same as Mode 1 on Reset)
OFF	OFF	MODE 3

Mode Selection by Software Control

The mode control circuitry of the PC-8011 can be controlled through software by executing an "OUT" instruction. The actual data transmitted has no effect. To change modes during program execution, execute an "OUT" instruction for the address of the mode you want. The addresses and the corresponding modes are listed in Table 2-2.

Table 2-2 Output Addresses and Corresponding Modes

OUTPUT ADDRESS	MODE
E0	MODE 0
E1	MODE 1
E2	MODE 2
E3	MODE 3

Table 3-1 RS-232-C Signal Lines (cont'd)

LINE NUMBER	SIGNAL	DESCRIPTION
6	Data Set Ready	This signal is sent to the PC-8011 by a peripheral device that is ready to receive data.
7	Signal Ground	
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	Data Terminal Ready	This signal is sent by the PC-8011 to peripheral devices when the PC-8011 is ready for transmission.
21		
22		

Table 3-1 RS-232-C Signal Lines (cont'd)

LINE NUMBER	SIGNAL	DESCRIPTION
23		
24		
25		
26		

Signal levels used in the RS-232-C interface are given in Table 3-2.

To perform serial-to-parallel conversions, the RS-232-C interface in the PC-8011 uses an NEC μPD8251. This LSI IC is a USART, which converts parallel 8-bit data sent to it by the CPU into a serial bit pattern for data transmission to a peripheral device and converts the serial bit pattern from the peripheral device to 8-bit parallel data for use by the CPU. The USART also provides the handshaking between the computer and its peripherals.

Table 3-2 RS-232-C Signal Levels

VOLTAGES		MIN	MAX
High Level Input Voltage	High Level Input Voltage	+5 V	+15V
	Low Level Output Voltage	-5 V	-15 V
Low Level Output Voltage	High Level Input Voltage	+6 V	+12 V
	Low Level Output Voltage	-6 V	-12 V

The 8251 is a programmable device with many operating modes. Before data can be transmitted or received, the 8251 must be initialized by software control to operate in the mode you want.

Because of its complexity, only those portions of the 8251 necessary for operation of the RS-232-C are explained here.

Before using the RS-232-C, you must set the baud rate clock supplied to the USART to match the baud rate of the peripheral device. Baud rate settings are described in the next section.

To initialize the 8251, load it with a mode instruction and a command instruction. The mode instruction specifies the 8251 basic transmission format and must be loaded immediately following an internal or external reset. (Internal resets are performed by command instructions.) Mode specification can be made only once following a reset. To change mode, perform a reset.

The command instruction controls the actual operation of the 8251 within the specified transmission format. You can change command instructions as often as necessary.

Perform these instructions by storing an 8-bit data word in the internal registers of the 8251 using N-Basic, as explained in the following section.

Data sent to the 8251 USART is interrupt-controlled. Each serial channel within the PC-8001 has its own 127-byte buffer and can receive data, at baud rates up to 4800, as long as the buffer is not full.

USING THE RS-232-C INTERFACE

Before using either of the serial I/O channels in the PC-8011, set the internal baud rate clock using DIP switch "SW2." Set the transmission rate for Channel 1 with switches 1 through 5 of SW2 (see Table 3-3). Set the transmission rate for Channel 2 with switches 6 through 10 of SW2 (see Table

3-4). These switches determine the frequency of the clock supplied to the USART. The actual baud rate is further determined by the operational mode, which can be programmed to X16 or X64.

Table 3-3 Channel 1 Baud Rates (Asynchronous Mode)

SW2*					BAUD RATE	
1	**2**	**3**	**4**	**5**	**X16**	**X64**
ON	OFF	OFF	OFF	OFF	4800	1200
OFF	ON	OFF	OFF	OFF	2400	600
OFF	OFF	ON	OFF	OFF	1200	300
OFF	OFF	OFF	ON	OFF	600	150
OFF	OFF	OFF	OFF	ON	300	75

*Do not turn on more than one switch at a time.

Table 3-4 Channel 2 Baud Rates (Asynchronous Mode)

SW2*					BAUD RATE	
6	**7**	**8**	**9**	**10**	**X16**	**X64**
ON	OFF	OFF	OFF	OFF	4800	1200
OFF	ON	OFF	OFF	OFF	2400	600
OFF	OFF	ON	OFF	OFF	1200	300
OFF	OFF	OFF	ON	OFF	600	150
OFF	OFF	OFF	OFF	ON	300	75

*Do not turn on more than one switch at a time.

The location of SW2 is shown in Figure 3-1.

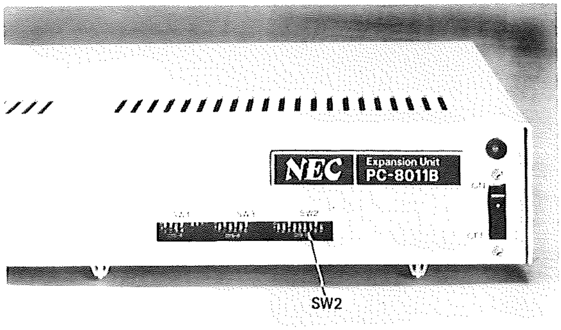


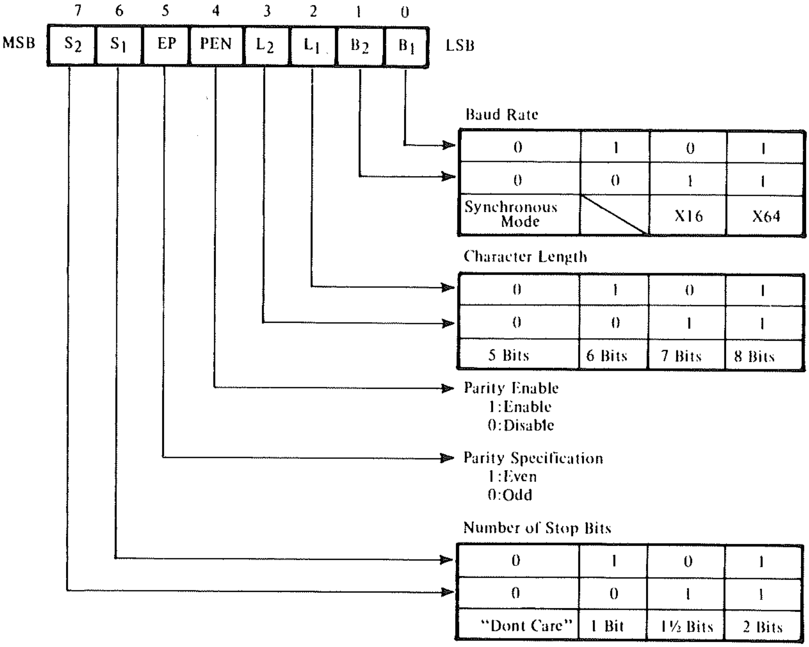
Figure 3-1 Switch SW2 Location

After you have set SW2, use N-Basic for the rest of the initialization. The normal power-on or reset sequence of the PC-8001 does not initialize the serial ports in the PC-8011. N-Basic has statements especially for this purpose, the OUT statement and the INIT statement. Their format is:

```
OUT    &HE4, &HFF
INIT   (PORT #), (Mode Type), (Command Byte)
```

The PC-8011 has only two serial ports. Therefore, (PORT #) can be 1 or 2. The second parameter of the INIT statement (Mode Type) is an 8-bit word that specifies the mode of the 8251, as previously explained. The format for the mode instruction is given in Figures 3-2 and 3-3. The INIT statement is necessary for both serial ports, but the OUT statement is not necessary for both serial ports. The third parameter of the INIT statement (Command Byte) specifies the command instruction, which has been previously explained. The format of the command byte, which is the same for synchronous and asynchronous modes, is given in Figure 3-4.

Figure 3-2 Asynchronous Mode Format



Baud Rate

0	1	0	1
0	0	1	1
Synchronous Mode		X16	X64

Character Length

0	1	0	1
0	0	1	1
5 Bits	6 Bits	7 Bits	8 Bits

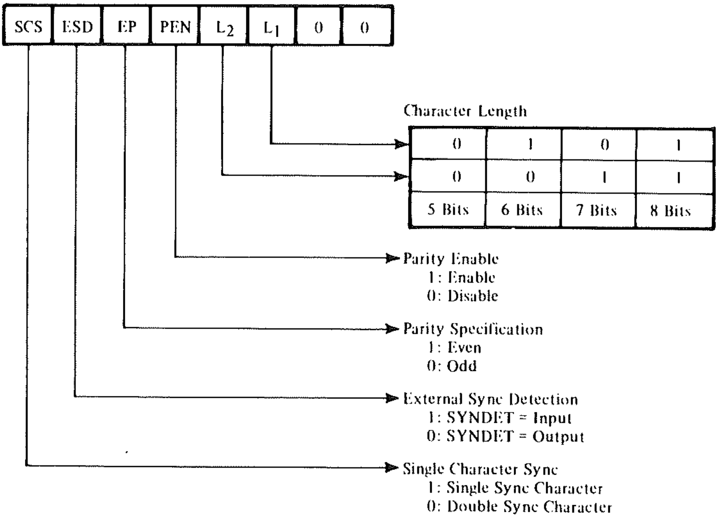
Parity Enable
1:Enable
0:Disable

Parity Specification
1:Even
0:Odd

Number of Stop Bits

0	1	0	1
0	0	1	1
"Dont Care"	1 Bit	1½ Bits	2 Bits

Figure 3-3 Synchronous Mode Format



Character Length

0	1	0	1
0	0	1	1
5 Bits	6 Bits	7 Bits	8 Bits

Parity Enable
1: Enable
0: Disable

Parity Specification
1: Even
0: Odd

External Sync Detection
1: SYNDET = Input
0: SYNDET = Output

Single Character Sync
1: Single Sync Character
0: Double Sync Character

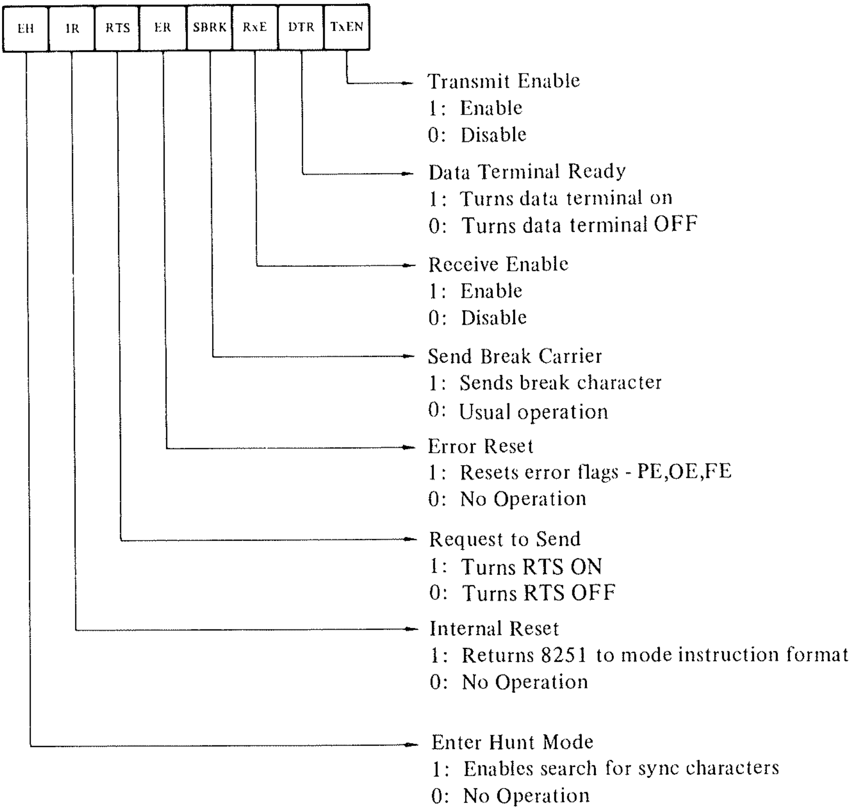


Figure 3-4 Command Byte

Example of INIT Statement:

```
INIT% 1, &HDA, &H15
```

After execution of this example, Channel 1 is configured as follows.

Mode:	Asynchronous
Baud Rate:	X16 mode
Character Length:	7 bits

Parity:	Odd
Number of Stop Bits:	2
State of Error Flags:	Reset
Transmit Enable:	Set
Receive Enable:	Set

You can change the command specification anytime after the INIT statement by using the OUT statement. The format of the OUT statement follows.

```
OUT (Port Address), (Command Byte)
```

The actual port addresses follow.

Port 1:	C1 (hex)
Port 2:	C3 (hex)

Example:

```
OUT &HC1, &H25
```

Execution of this example sets the RTS bit high.

The status of the 8251 USART can be read using the INP function.

Example:

```
A = INP (Port Address)
```

Or, if you want, the answer can be returned in hex.

Example:

```
A$ = Hex$ (INP (Port Address))
```

The port address is the same as explained in the OUT statement.

Example:

```
A      = inp(&HC1):printA
133
OK
A$     = hex$(INP(&HC1)):PrintA$
85
OK
```

The actual format of the status word is given in Figure 3-5.

Data can be sent to an RS-232-C channel with a Print% statement. The format for this statement follows:

Print% Port #, output data list

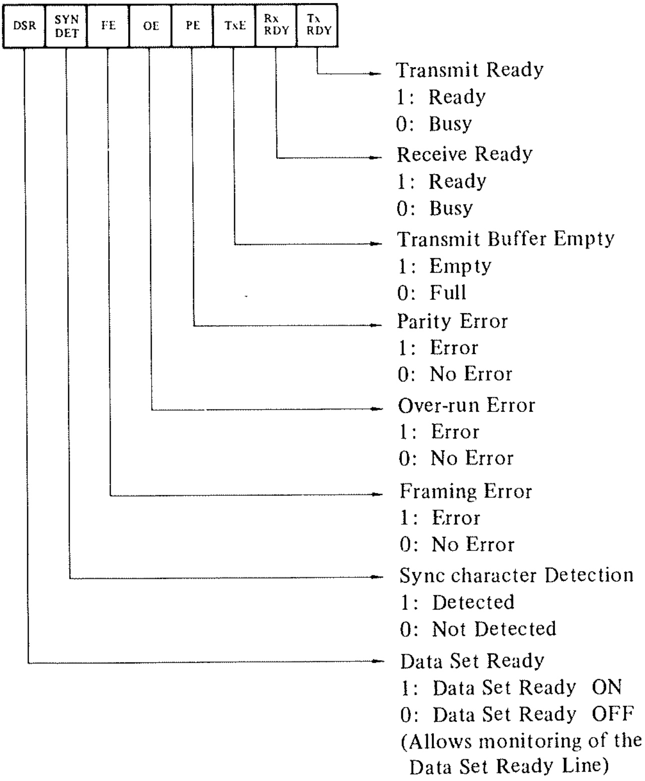


Figure 3-5 Status Word

Example 1:

```
Print% 1, a, b, c, d
```

When this statement is executed, the values of variables a, b, c, and d are sent over Channel 1.

Example 2:

```
Print% 2, "EXAMPLE", A$
```

When Example 2 is executed, the values of the string constant "EXAMPLE" and the string variable A$ are sent over Channel 2.

As previously explained, the RS-232-C interface of the PC-8011 operates under interrupt control, with incoming data stored temporarily as needed in a buffer. Data in the input buffer can be read by assigning it to string variables using the INPUT% statement. The format for INPUT% follows.

```
INPUT% Port #, Variable List
```

In this case, the delimiter is a carriage return followed by a line feed. All data between delimiters is regarded as one string and assigned to the corresponding variable. If a delimiter is not found in the buffer, execution waits until a delimiter is sent in over the RS-232-C interface.

Example:

```
INPUT% 1, a$, B$, C$:  PRINT a$, B$, C$
ABCDEF
12345678
NEC
OK
```

NOTE

In this case, these three items of data must be terminated by ",". ABCDEF, 12345678, NEC, CR, L/F.

Using the INPUT$ function, strings of a specified length can be sent from an RS-232-C input buffer and assigned to a specified variable. The format for this function follows.

```
INPUT$ (string length, % Port #)
```

Example:

```
A$ = input$(3,%1):PrintA$
```

When this example is executed, three characters are read over Channel 1 and assigned to the variable A$.

The PORT function returns the number of characters in the input buffer of the specified port. The format of the PORT function follows.

```
PORT (Port #)
```

Example:

```
A = PORT (1): Print A
23
OK
```

Two error messages concern the RS-232-C interface:

- **Port Not Initialized** — Correct this problem by initializing the 8251 with the INIT% statement.

- **Communication Buffer Overflow** — This error message indicates that an attempt was made to enter data when the input buffer was full.

Example Program:

```
100 '
110 '**********************************
120 ' PC-8011 RS-232 channel demo
130 '     no SI/SO insertion is provided
140 '**********************************
150 '
160 ' set up
170 OUT &HE4, &HFF
180 LINEINPUT "Want hard copy?";A$
190 IF A$="y" THEN COPY=1 ELSE COPY=0
200 LINEINPUT "Want lower case?";A$
210 IF A$="y" THEN LOWER=1 ELSE LOWER=0
220 LINEINPUT "Want self echo?";A$
230 IF A$="y" THEN ECHO=1 ELSE ECHO=0
240 LINEINPUT "Want auto L/F?";A$
250 IF A$="y" THEN LF=1 ELSE LF=0
260 '
270 ' Initialze uPD8251 as follows:
280 '  2 stop bits
290 '  no parity
300 '  8 bit data
310 '  *16 baud rate factor
320 '  RTS=1,DTR=1
330 '
340 INIT %1,&HCE,&H37
350 '
360 ' test queue and read if any,
370 ' then print it
380 '
390 IF PORT(1) THEN A$=INPUT$(1,%1):GOSUB 530:PRINT A$;:IF COPY THEN LPRINT A$;
400 '
410 ' test console status.
420 ' read also if any,and xmit it
430 '
440 A$=INKEY$:IF A$<>"" THEN GOSUB 530:PRINT %1,A$;:GOSUB 460:IF ECHO THEN PRINT
 A$;:IF COPY THEN LPRINT A$;
450 GOTO 390
460 '
470 ' append a L/F
480 ' if a character is C/R,
490 ' and L/F switch is on.
500 '
510 IF LF THEN IF A$=CHR$(13) THEN A$=A$+CHR$(10)
520 RETURN
530 '
540 ' convert lower case letter
550 ' into upper case,
560 ' if LOWER case switch is off
570 '
580 IF LOWER THEN RETURN
590 IF A$>="a" AND A$<="z" THEN A$=CHR$(ASC(A$)-32)
600 RETURN
```

CHAPTER 4

I/O PORTS

The PC-8011 has an I/O port for the controller in the PC-8031 Mini-Disk Drive Unit. When the PC-8011 Expansion Unit is used, the PC-8033 FDC I/O unit is not necessary.

Interconnect the two units with the 34-conductor flat cable provided with the PC-8031 (see Figure 4-1) so that the "Δ" mark on the case of the PC-8011 is in line with the colored conductor of the flat cable.

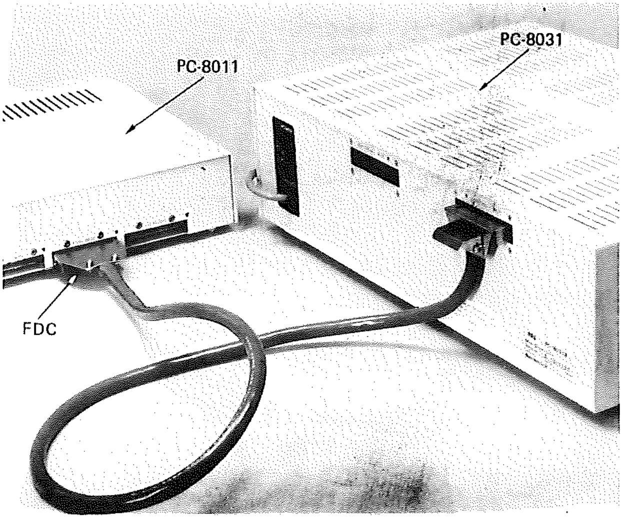


Figure 4-1 PC-8011 Connected to PC-8031 Mini-Disk Drive Unit

PARALLEL I/O PORTS

The PC-8011 has several general purpose parallel I/O ports. All I/O signal levels for parallel ports are TTL levels. You can configure parallel I/O ports and use them for data transmission with programs written in machine language or N-Basic.

Eight-Bit Input Port

The I/O address of the 8-bit input port is B0 (Hex). Positive logic data presented at this port can be read onto the data bus.

Eight-Bit Output Port

The I/O address of the 8-bit output port is B1 (Hex). Data sent to address B1 is presented at this port as positive logic data.

Four-Bit Input Port

The I/O address of the 4-bit input port is B2 (Hex). This port uses positive logic and sets its data on the lower four bits of the data bus.

Four-Bit Output Port

The I/O address of the 4-bit output port is B3 (Hex). When the PC-8001 is reset, all lines of this port go high. Data on the lower four lines of the data bus is inverted and sent to this port as negative logic data.

PARALLEL I/O PORT SIGNALS

All parallel I/O port signals are TTL-level with a fan-out of ten LS TTL loads. These signals and their associated terminal are shown in Table 4-1.

Table 4-1 Parallel I/O Port Signals

TERMINAL	SIGNAL	TERMINAL	SIGNAL
1	GND	2	DO0
3	GND	4	DO1
5	GND	6	DO2
7	GND	8	DO3
9	GND	10	DO4

Table 4-1 Parallel I/O Port Signals (cont'd)

TERMINAL	SIGNAL	TERMINAL	SIGNAL
11	GND	12	DO5
13	$\overline{\text{INT9}}$	14	DO6
15	$\overline{\text{INT8}}$	16	DO7
17	IN0	18	DI0
19	IN1	20	DI1
21	IN2	22	DI2
23	IN3	24	DI3
25	$\overline{\text{OUT0}}$	26	DI4
27	$\overline{\text{OUT1}}$	28	DI5
29	$\overline{\text{OUT2}}$	30	DI6
31	$\overline{\text{OUT3}}$	32	DI7
33	+5	34	GND

GND

These lines serve as signal ground for all other signals.

$\overline{\text{INT8}}$, $\overline{\text{INT9}}$

These are interrupt lines. For details, see Chapter 7.

IN0 — IN3

These lines carry the data for the 4-bit input port.

OUT0 — OUT3

These lines carry the data from the 4-bit output port.

DO0 — DO7

These lines comprise the 8-bit output port.

DI0 — DI7

These lines make up the 8-bit input port.

Timing charts for these signals are given in Figures 4-2 through 4-4.

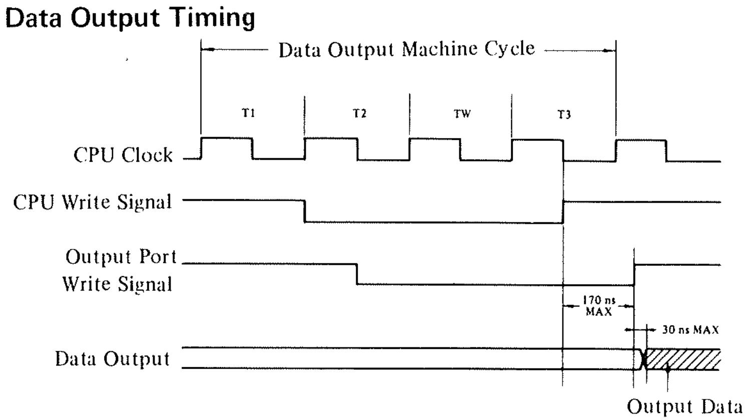


Figure 4-2 Data Output Timing

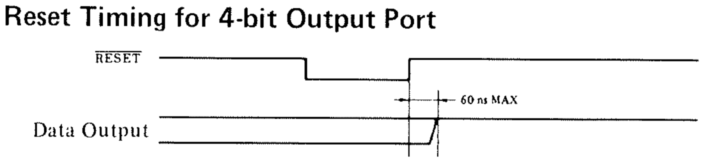


Figure 4-3 Reset Timing for Four-Bit Output Port

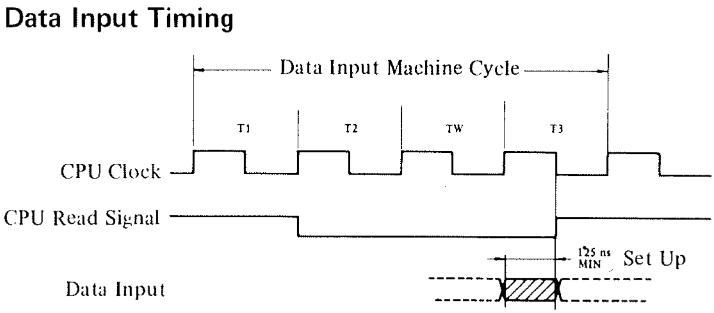


Figure 4-4 Data Input Timing

USING THE PARALLEL I/O PORTS

The parallel I/O ports (see Figure 4-5) can be programmed for a number of configurations. The explanation given here concerns only the most common configurations.

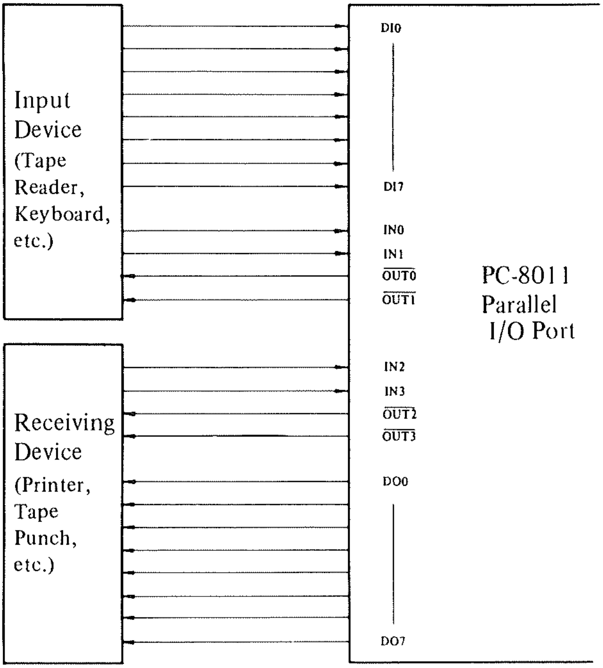


Figure 4-5 Parallel I/O Port Configuration

Normally the computer and I/O device must exchange some kind of control signals. This process is called "handshaking," where the computer and I/O device exchange such information as status reports, read timing, and start timing. In the PC-8011, these signals are passed over the two 4-bit I/O ports. Input devices use lines DI0 to DI7, and devices being transmitted use lines DO0 to DO7.

The parallel I/O port of the PC-8011 is a general-use port, and the data lines can be used in any configuration.

Application Example 1: Connecting a Keyboard

A wiring diagram for connecting a keyboard to the parallel port is shown in Figure 4-6. The following program controls the circuit given in Figure 4-6. The program waits until you press one of the keys, 0 to 15, then it reads the value of that key and assigns it to the variable F.

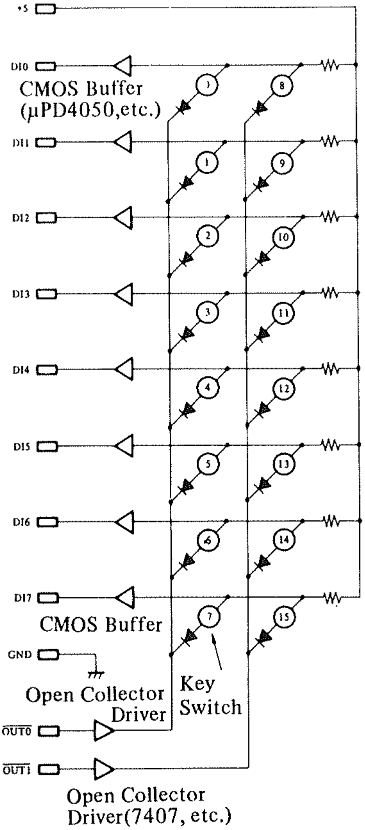


Figure 4-6 Keyboard Circuit

Application Example 2: Connecting a Paper Tape Reader

The circuit for connecting DPF-6A paper tape reader to the parallel port is given in Figure 4-7. The following program reads in one character and assigns that character to the variable B$.

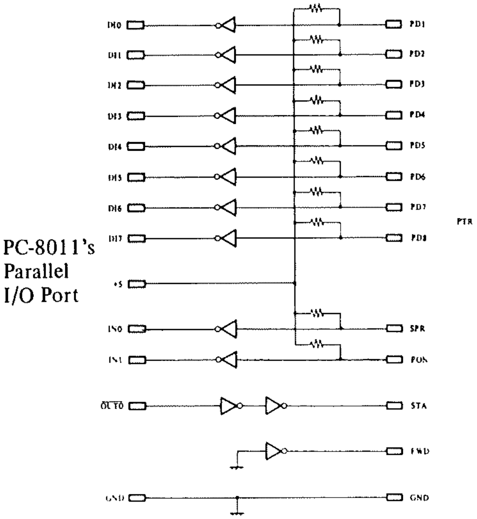


Figure 4-7 Paper Tape Reader Circuit Connections

```
10 DATA &HF5, &HDB, &HB2, &HE6, &H02, &HCA
   &H81, &HFF
20 DATA &H3D, &HD3, &HB3, &HDB, &HB2, &HE6,
   &H01, &HCA
30 DATA &H8B, &HFF, &HDB, &HB0, &H32, &HA0,
   &HFF, &H97
40 DATA & HD3, &HB3, &HF1, &HC9
50 FOR A=&HFF80 to &HFF9B
60 READ B:  POKE A,B:  NEXT A
70 DEFUSR 0=&HFF80
```

Character-read routine from PTR

```
1000 A =USR0(1)
1010 B=PEEK(&HFFA0)
1020 B$=CHR$(B)
1030 RETURN
```

Application Example 3: Connecting a Printer

The general circuit for connecting a printer to the parallel port is given in Figure 4-8. The following program sends the first character of the string assigned to the string variable A$.

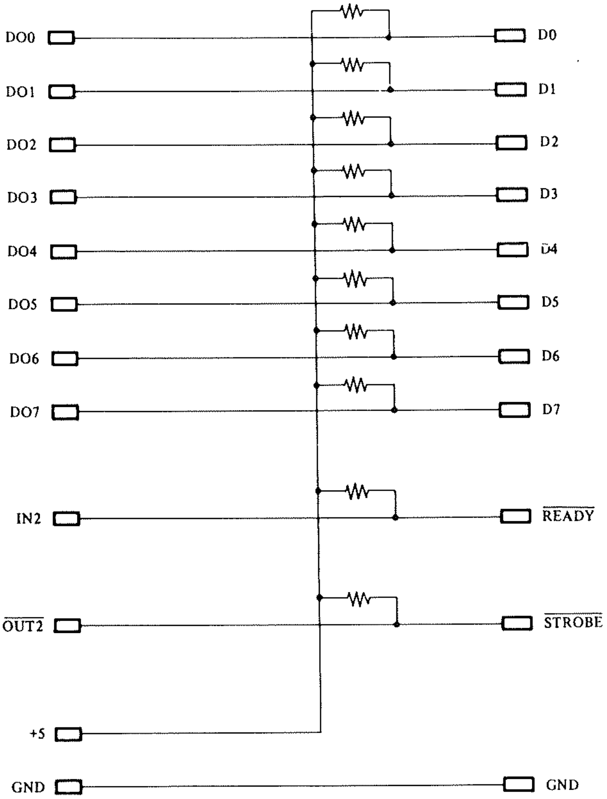
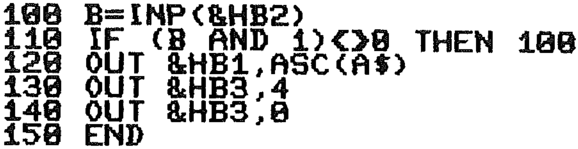


Figure 4-8 Printer Circuit Connections



```
100 B=INP(&HB2)
110 IF (B AND 1)<>0 THEN 100
120 OUT &HB1,ASC(A$)
130 OUT &HB3,4
140 OUT &HB3,0
150 END
```

The above program outputs the first character of the string assigned to the string variable A$.

CHAPTER 5

IEEE-488 INTERFACE

The PC-8011 contains an IEEE-488 interface as standard equipment. Any device using this bus can be connected directly to the PC-8011. The individual signal lines are shown in Figure 5-1.

DATA BUS	DIO1	HANDSHAKING SIGNALS	DAV
	DIO2		NRFD
	DIO3		NDAC
	DIO4	BUS CONTROL SIGNALS	IFC
	DIO5		ATN
	DIO6		EOI
	DIO7		SRQ
	DIO8		REN

Figure 5-1 IEEE-488 Signal Lines

Lines DIO1 through DIO8 form an 8-bit directional data bus. Lines DAV, NFRD, and NDAC perform handshaking. Lines IFC, ATN, EOI, SRQ, and REN carry signals used by the bus controller.

The functions of the IEEE-488 interface of the PC-8011 are given in Table 5-1.

Table 5-1 Functions of the IEEE-488 Interface

FUNCTION	SUBSET	FUNCTION
SH	SH1	Complete Function
AH	AH1	Complete Function
T	T6	Basic Talker Serial Poll Cancel Talker by MTA
L	L4	Basic Listener Cancel Listener by MTA
SR	SR1	Complete Function
RL	RL0	No RL Function
PP	PP0	No PP Function
DC	DC0	No DC Function
DT	DT0	No DT Function
C	C1 C2 C3 C4 C28	System Controller IFC, REN Transmission Reply to SRQ Transmission of Interface Messages

The software for using the IEEE-488 bus is not a standard feature of N-Basic and must be added by installing an optional ROM IC. For details on installation of this IC, see the manual provided with it. The software adds the following statements to N-Basic.

ON SRQ GOSUB

Execution of this statement causes the interpreter to jump to the specified line number on the next service request (SRQ).

POLL

This statement performs a serial poll of the device at the specified address.

ISET

Activates the REN or IFC signal

IREST REN

Resets the REN signal.

PRINT@

Sends a character string to the device at the specified address.

INPUT@

Assigns the data received by the device at the specified address as the value of a character string.

LINE INPUT@

Reads the data received by the device at the specified address up to the first character return and line feed, and assigns it to a character string.

WBYTE

Sends data, such as multiline messages, to the interface bus.

RBYTE

Reads the data received by the device at the specified address and assigns it as the value of a numeric variable.

STATUS

Presets the status byte when making service requests.

IEEE Function

Returns bus status.

CMD DELIM

Specifies a delimiter.

CHAPTER 6

EXTENDED I/O BUS

The PC-8011 has provisions for extending the I/O bus.

All data, control, and interrupt lines that make up the extended I/O bus include drivers. The I/O addresses available on the extended I/O bus are 80 through AF (hex). Memory cannot be connected to the I/O bus.

Interrupts to the PC-8001 can be made using $\overline{\text{INT0}}$ through $\overline{\text{INT7}}$. When an interrupt occurs, the vector data generated during that acknowledge cycle that corresponds to the interrupt is set on the data bus. By reading this data, the system can tell which device is requesting an interrupt. Interrupt vector data is given in Table 6-1.

Table 6-1 Interrupt Vector Data

Priority Order	Interrupt Line	Vector Data	
		MSB Binary LSB	Hex
Highest	$\overline{\text{INT9}}$	0 0 0 0 1 1 0 0	0C
	$\overline{\text{INT8}}$	0 0 0 0 1 1 1 0	0E
	$\overline{\text{INT7}}$	0 0 0 1 0 0 0 0	10
	$\overline{\text{INT6}}$	0 0 0 1 0 0 1 0	12
	$\overline{\text{INT5}}$	0 0 0 1 0 1 0 0	14
	$\overline{\text{INT4}}$	0 0 0 1 0 1 1 0	16
	$\overline{\text{INT3}}$	0 0 0 1 1 0 0 0	18
	$\overline{\text{INT2}}$	0 0 0 1 1 0 1 0	1A
	$\overline{\text{INT1}}$	0 0 0 1 1 1 0 0	1C
Lowest	$\overline{\text{INT0}}$	0 0 0 1 1 1 1 0	1E

NOTE: Lines $\overline{\text{INT8}}$ and $\overline{\text{INT9}}$ are also included in the I/O Port Bus. For details on interrupts, see Chapter 7.

The extended I/O bus signals are given in Table 6-2.

Table 6-2 Extended I/O Bus Signals

Term No.	Signal	Term No.	Signal
1	$\overline{\text{INT0}}$	2	IDB0
3	$\overline{\text{INT1}}$	4	IDB1
5	$\overline{\text{INT2}}$	6	IDB2
7	$\overline{\text{INT3}}$	8	IDB3
9	$\overline{\text{INT4}}$	10	IDB4
11	$\overline{\text{INT5}}$	12	IDB5
13	$\overline{\text{INT6}}$	14	IDB6
15	$\overline{\text{INT7}}$	16	IDB7
17	GND	18	$\overline{\text{IOR}}$
19	GND	20	$\overline{\text{IOW}}$
21	GND	22	A0
23	GND	24	A1
25	GND	26	A2
27	GND	28	A3
29	GND	30	$\overline{\text{EXT0}}$
31	GND	32	$\overline{\text{EXT1}}$
33	GND	34	$\overline{\text{EXT2}}$
35	GND	36	$\overline{\text{RESET}}$
37	GND	38	$\overline{\text{INTV}}$
39	GND	40	$\overline{\text{WAIT}}$
41	GND	42	$\overline{\text{NMI}}$
43	GND	44	$\overline{\text{M1}}$
45	+12	46	SCLOCK
47	–12	48	ϕ
49	+5	50	GND

$\overline{\text{INT0}}$ — $\overline{\text{INT7}}$

These lines are active low for sending interrupts to the PC-8001 and generating interrupt vectors. For details, see Chapter 7.

IDB0 — IDB7

These lines form a data bus. I/O addresses available for this bus are 80 to AF (hex).

IOR

This I/O Read signal goes low when the $\overline{\text{RD}}$ signal of the CPU is active during execution of an IN instruction for an I/O address from 80 to AF (hex).

$\overline{\text{IOW}}$

This I/O Write signal goes low when the $\overline{\text{WR}}$ signal of the CPU is active during execution of an OUT instruction for an I/O address from 80 to AF (hex). Data is latched at the output port on the positive edge of this signal.

A0 — A3

These signal lines are the lower four lines of the address bus of the CPU and are decoded and used along with lines $\overline{\text{EXT0}}$ through $\overline{\text{EXT2}}$ to select peripherals.

$\overline{\text{EXT0}}$ — $\overline{\text{EXT2}}$

These lines are the result of decoding the upper four bits of the I/O address (8 bits). The ranges of addresses that they correspond with are listed in Table 6-3.

Table 6-3 $\overline{\text{EXT0}}$ — $\overline{\text{EXT2}}$ Ranges of Addresses

Signal	I/O Address
$\overline{\text{EXT0}}$	80 — 8F
$\overline{\text{EXT1}}$	90 — 9F
$\overline{\text{EXT2}}$	A0 — AF

These signals are used along with A0 through A3 to select peripherals.

RESET

This signal comes directly from the bus of the PC-8001. When the PC-8001 is reset, $\overline{\text{RESET}}$ goes low for approximately 20 μs.

$\overline{\text{INTV}}$

This signal is the interrupt acknowledge signal. It goes low during the interrupt acknowledge cycle. Data set on the data bus, when this signal goes active after an interrupt has been received, is used to clear the interrupt.

$\overline{\text{WAIT}}$

The CPU of the PC-8001 can be put in a wait state by setting this line low.

> **CAUTION**
>
> Both the PC-8001 and PC-8011 use dynamic RAM, which must be continually refreshed by the CPU. Refresh, which is normally performed automatically, does not occur during wait states. If the CPU is held in a wait state too long, the contents of RAM can be lost.

$\overline{\text{NMI}}$

This line is used for sending nonmaskable interrupts to the CPU. $\overline{\text{NMI}}$ is active low.

$\overline{\text{MI}}$

This signal is active low, indicating that the CPU is in machine cycle one.

SCLOCK

This signal has a frequency of 76.8 kHz and a 50% duty cycle. It can be used as a baud rate clock.

ϕ

This is the CPU clock.

Timing for the signals on the extended I/O bus is shown in Figure 6-1. All signals are TTL level. Output signals drive ten standard LS TTL loads.

The actual time in nanoseconds for components of the signals is given in Table 6-4.

Table 6-4 Signal Components Time

SIGNALS	MIN	TYP	MAX	TIME
T_{RR}	550			ns
T_{AR}	80			ns
T_{RA}	0			ns
T_{CR}			50	ns
T_{RC}	0			ns
T_{OR}	270			ns
T_{RD}	0			ns
T_{WTR}	70			ns
T_{TWR}	375			ns
T_{WW}	400			ns
T_{AW}	200			ns
T_{WA}	0			ns
T_{CW}	20			ns

Table 6-4 Signal Components Time (cont'd)

SIGNALS	MIN	TYP	MAX	TIME
T_{WC}	0			ns
T_{DW}	20			ns
T_{WD}	0			ns
T_{WTW}	20			ns
T_{TWW}	250			ns
T_{INT}	0			ns
T_{ID}	30			ns
T_{DI}	0			ns
T_{WI}	65			ns
T_{RESST}	10			μs

USING THE EXTENDED I/O BUS

This section explains the basic procedures for connecting peripherals to the extended I/O bus. Peripherals are selected by decoding lines $\overline{\text{EXT0}}$ to $\overline{\text{EXT2}}$ and lines A0 to A3. The $\overline{\text{IOR}}$ signal can be used directly as the read strobe, and the $\overline{\text{LOW}}$ signal can be used directly as the write strobe. The circuits for connecting various ports are given in Figures 6-2, 6-3, and 6-4.

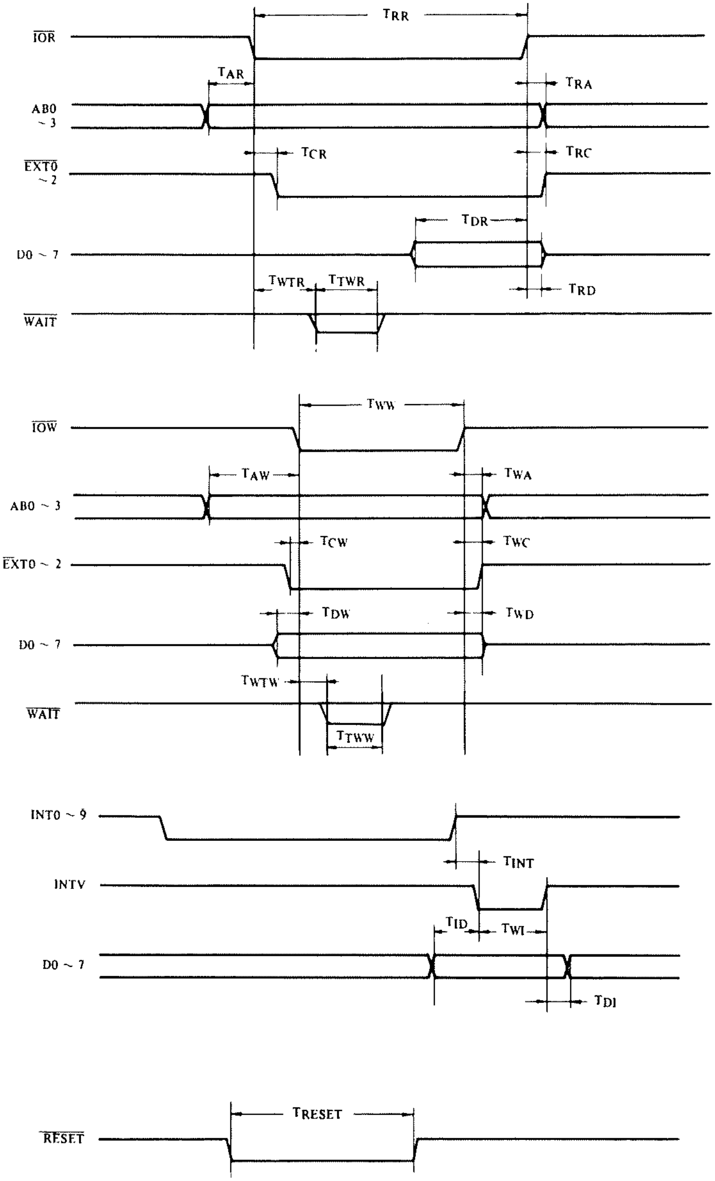
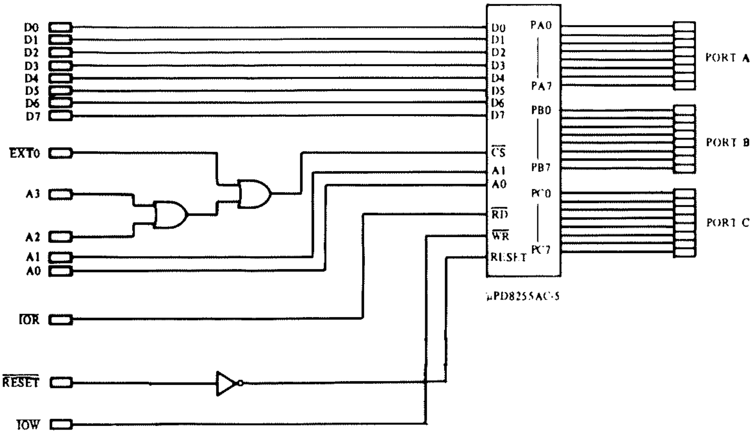
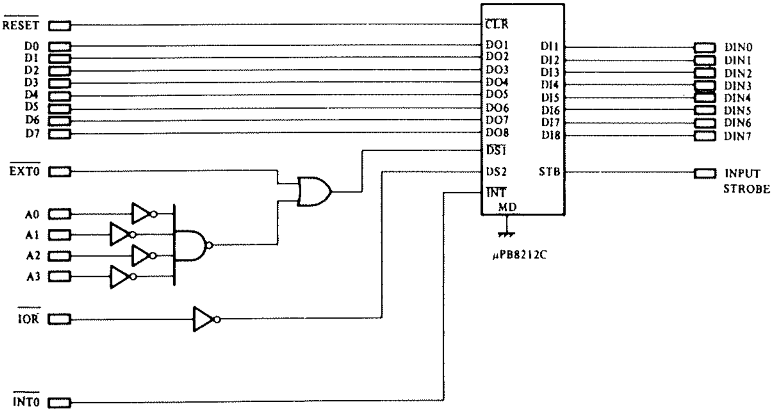


Figure 6-1 Timing Signals



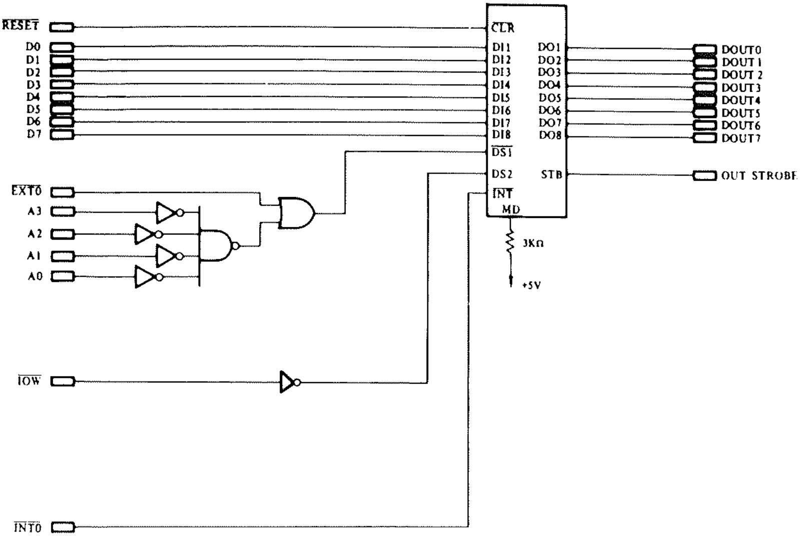
Circuit for connecting a PPI(μPD8255AC-5)
I/O Address 80 (16 Hex) . . . Port – A
81 (16 Hex) . . . Port – B
82 (16 Hex) . . . Port – C
83 (16 Hex) . . . Control

Figure 6-2 Circuit for Connecting a μPD8255AC-5



Circuit for connecting interrupt input port
(μPD8212C) (IN Address=80 Hex)

Figure 6-3 Circuit for Connecting an Interrupt Input Port



Circuit for connecting interrupt output port
(μPD8212C) (OUT Address=80 Hex)

Figure 6-4 Circuit for Connecting an Interrupt Output Port

CHAPTER 7

INTERRUPTS

Interrupt control logic is a standard feature of the PC-8011. The CPU (μPD780C-1) runs in Mode 2 as determined by the system program, N-Basic. The PC-8001 has 16 interrupt priority levels. The system uses five, and one level is unused; ten levels are available to you.

MODE 2 INTERRUPT

The μPD780C has provisions for both maskable and non-maskable interrupts. The PC-8011 can use both types of interrupts. However, only maskable interrupts can be used while N-Basic is running.

Maskable interrupts, hereafter INT, can be enabled or disabled under software control. Thus, even if interrupts are being used by a program, continuous execution of important routines can be assured by setting the interrupt mask at the head of the routine. This is done with the interrupt mask flag ("IFF"), which can be set and cleared by program control.

The μPD780C has three interrupt response modes, any of which can be selected by software control. When N-Basic is running, Mode 2 is selected.

In Mode 2, the programmer must maintain a table of 16-bit addresses that serve as interrupt vectors somewhere in memory (see Figure 7-1). When an interrupt occurs, a 16-bit pointer must be formed to get the starting address of the routine you want from this table. The upper 8 bits of this pointer are formed from the contents of the I register, and the lower 8 bits must be supplied by the interrupting device to the CPU during the interrupt acknowledge cycle.

STRUCTURE OF PC-8011 INTERRUPT TABLE

Interrupt logic in the PC-8011 uses two interrupt priority encoders (μPB8214X) to handle 16 levels of interrupts.

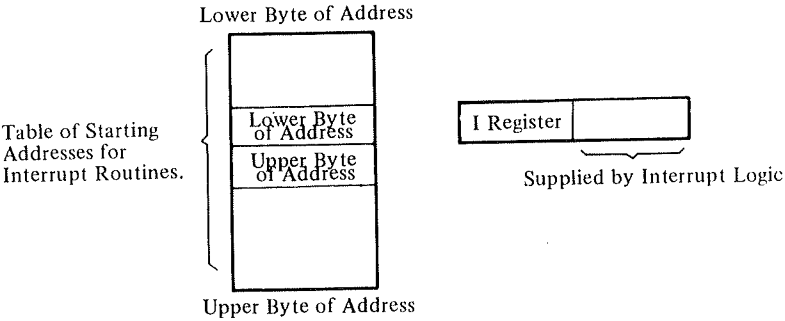


Figure 7-1 Interrupt Vector Configuration

The interrupt table is located in the work area within the RAM of the PC-8001. This table is located from 0000 to 001F in 16K systems, and from 8000 to 802F in 32K systems. The correspondence between interrupt channels and locations within the interrupt table is given in Table 7-1.

Table 7-1 Interrupt Channels and Addresses

Interrupt Table Addresses	**Content**	**Channel**
8000 1	Used by N-BASIC	IEEE-488
2 3	Used by N-BASIC	IEEE-488
4 5	Low-order byte High-order byte	Real-Time Clock
6 7		
8 9	Used by N-BASIC	RS-232C Ch. 1
A B	Used by N-BASIC	RS-232C Ch. 2

Table 7-1 Interrupt Channels and Addresses (cont'd)

Interrupt Table Addresses	Content	Channel
C D	Low-order byte High-order byte	$\overline{\text{INT9}}$
E F	Low-order byte High-order byte	$\overline{\text{INT8}}$
10 11	Low-order byte High-order byte	$\overline{\text{INT7}}$
12 13	Low-order byte High-order byte	$\overline{\text{INT6}}$
14 15	Low-order byte High-order byte	$\overline{\text{INT5}}$
16 17	Low-order byte High-order byte	$\overline{\text{INT4}}$
18 19	Low-order byte High-order byte	$\overline{\text{INT3}}$
1A 1B	Low-order byte High-order byte	$\overline{\text{INT2}}$
1C 1D	Low-order byte High-order byte	$\overline{\text{INT1}}$
1E 801F	Low-order byte High-order byte	$\overline{\text{INT0}}$

USING THE INTERRUPT CONTROLLER

The PC-8011 uses the μPB8214C interrupt priority controller to handle multilevel interrupts and to generate pointers for the interrupt table.

The 8214 can be roughly divided into two functional sections. One section consists of the priority encoder, which assigns priority levels. The other section is the current status register section, which is used to mask interrupts. One 8214 can handle eight levels of interrupts. The PC-8011 uses two cascaded 8214s to handle 16 interrupt levels.

The priority encoder of the 8214 handles interrupt request signals for eight channels. The relative priorities of these eight channels are determined by hardware configuration. When the priority encoder receives two or more interrupt requests, it chooses the interrupt with the highest priority and encodes the priority level into a 3-bit value. The priority encoder has an interrupt signal latch, which is controlled by the interrupt inhibit flip-flop.

Once processing of an interrupt by the CPU begins, the current status section masks any incoming interrupts of lower priority. For details, see Appendix C.

USING INTERRUPTS

Using interrupts requires that you understand the associated hardware and how the machine processes interrupts.

Hardware

An example of the hardware necessary for using the interrupt facilities of the PC-8011 is given in Figure 7-1. The circuit shown latches interrupt request signals until they can be received by the CPU. Interrupt requests transmitted from external devices over the active-low signal lines $\overline{\text{INTRQ0}}$ through $\overline{\text{INTRQ9}}$ set the SR flip-flops, informing the system that an interrupt has occurred. A single pulse over an internal interrupt line is all that is needed to make an interrupt request.

When the SR flip-flops are set, their $\overline{\text{Q}}$ output, which is connected to lines $\overline{\text{INT0}}$ through $\overline{\text{INT9}}$ of extended I/O bus, go low. These signal lines are directly connected to 8214s within the PC-8011. When the $\overline{\text{INTV}}$ signal goes low during the acknowledge cycle after the CPU has received an interrupt request, data specifying the priority level of the interrupt is set onto lines D1 through D4 of the extended I/O bus. The data on these four lines is decoded to reset the flip-flop of the interrupt that has just been received. The circuitry with the SR flip-flops is shown in Figure 7-2.

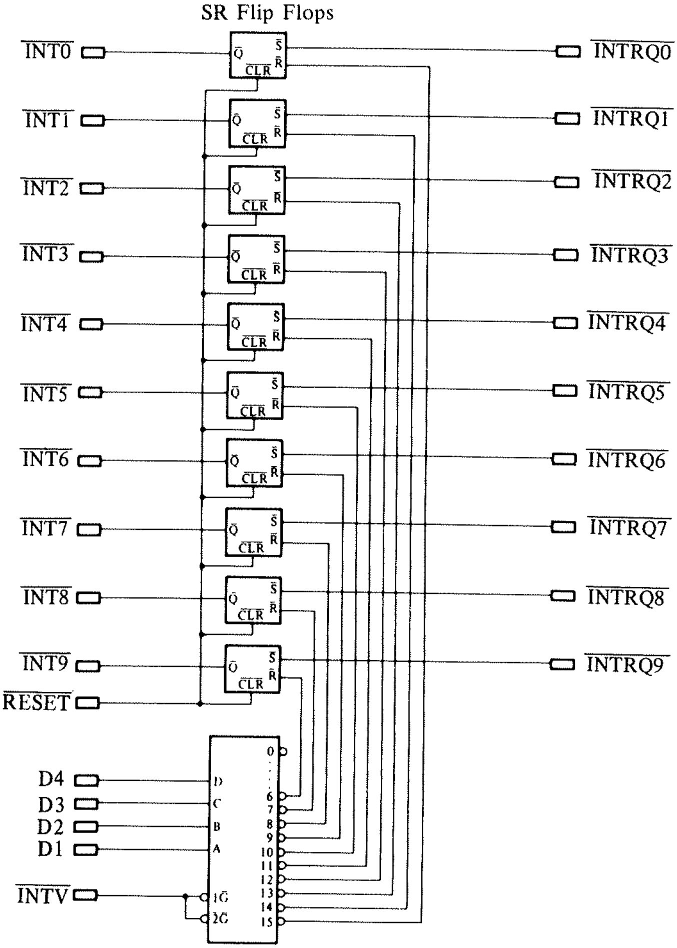


Figure 7-2 SR Flip-Flop Circuitry

Processing Interrupts

Using interrupts with the PC-8011 differs from normal interrupt processing in that the current status register of the 8214 must be operated and the interrupt flag must be set. A generalized program for this procedure follows.

```
(Entry point for interrupt routine)

(save registers)
     LD                        A,(0EA55H)             (0)
     PUSH                      AF
     (interrupt processing)        (DI)               (1)
     LD                        A, (interrupt label)   (2)
     OUT                       (8214   address),A     (3)
     LD                        (0EA55H),A             (4)
     LD                        (0EA56H),A             (5)
     EI
     (interrupt processing)        (EI)               (6)
     POP                       AF
     LD                        (0EA55H),A             (7)
     OUT                       (8214 address),A
     (restore registers)
     RET                                              (8)
```

An explanation of the numbered steps of this program follows.

(0) The priority level of the current interrupt; that is, the one immediately before the latest interrupt to occur, is stored at address 0EA55H.

(1) The actual processing required by the interrupt is started in this step and is performed in the interrupt disabled state. An example of such processing would be the reading of the receive buffer when the RXrdy line of the 8251 goes active, indicating that a character has been entered.

(2) and (3) The priority level of the interrupt is written into the current status register of the 8214. Interrupt levels and the addresses of the 8214 to which they correspond are given in Table 7-2.

Table 7-2 Interrupt Levels and Addresses

Channel	Interrupt Level	8214 Address
$\overline{\text{INT9}}$	6	0E4H
8	7	0E4H
7	0	0E5H
6	1	
$\overline{\text{INT5}}$	2	
4	3	
3	4	
2	5	
1	6	
0	7	

(4) The level of the new interrupt is stored at address EA55H.

(5) The flag indicating that an interrupt has occurred is set by storing any value other than zero at address E456H.

(6) This processing is any interruptable processing within the interrupt routine. (There may be none.)

(7) All registers saved in step (0) are stored.

(8) A return is made to the routine originally interrupted.

Example Program

The following program will read in one character from channel 1 of the RS-232-C interface and will store it at address "XYZ".

entry point

```
PUSH        AF
```

```
LD      A,(0EA55H)
PUSH    AF
IN      0C0H
LD      (XYZ),A
LD      A,4
OUT     0E4H,A
LD      (0EA55H),A
LD      (0EA56H),A
EI
POP     AF
LD      (0EA55H),A
OUT     0E4H,A
POP     AF
RET
```

CHAPTER 8

REAL-TIME INTERRUPTS

The PC-8011 has facilities for real-time interrupt processing to service peripheral devices at regular intervals.

MEANING OF REAL-TIME INTERRUPT

The real-time clock used by the PC-8011 is derived by dividing the system clock of the PC-8001 down to 600 Hz. This clock is supplied to the interrupt control logic within the PC-8011, which uses it to interrupt the CPU every 1.67 ms.

The real-time interrupt facility is especially convenient when a large number of external devices must be serviced and when high-speed replies are not necessary. Normally, peripheral equipment is serviced by using interrupts. However, there are only ten interrupt channels open to you. For those cases where high-speed responses are not required, the real-time interrupt facility can be used, along with software, for the most efficient use of interrupt channels.

By reading input ports, which are connected to service request lines of peripheral equipment, the CPU can tell which device is requesting service. The real-time input facility can be used to read the input ports at fixed intervals, thus providing regular service to peripheral devices. With this arrangement, the CPU services a number of devices with only one interrupt channel. However, there is some additional overhead involved with reading the I/O lines. Response time is determined by the frequency of the real-time clock. A block diagram and logic diagram of real-time interrupts are shown in Figure 8-1.

GENERALIZED REAL-TIME PROGRAM

A generalized program for real-time interrupt procedures follows.

(Entry point for interrupt routine)

1	1	1	1	1	1	0	0	Port A (Handshaking)
1	1	1	1	1	1	0	1	Port B (Handshaking)
1	1	1	1	1	1	1	0	Port C (Handshaking)
1	1	1	1	1	1	1	1	Control Port (Handshaking)

Figure A-7 Disk Control

APPENDIX B

CABLES AND CONNECTORS

The cable for connecting the PC-8011 and PC-8001 is included as standard equipment.

All of the PC-8011 I/O signals are available on edge connectors on the main PCB. The following cables and connectors or their equivalent should be used for making connection with these edge connectors (see Figure B-1).

INTERFACE OR PORT	PART NO. OF CABLE
RS-232C	PC-8095
IEEE-488	PC-8096
Parallel I/O port	3M 3463-0001
Extended I/O bus	3M 3415-0001

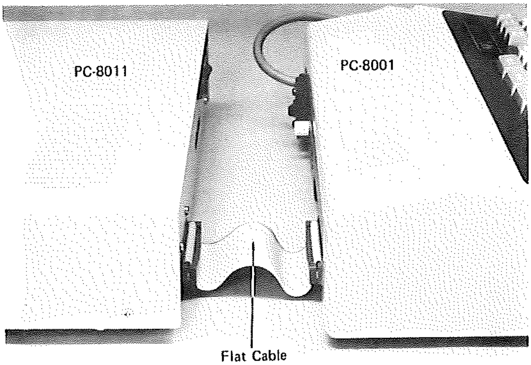


Figure B-1 PC-8011 Connecting Cable

NOTE

Use shielded flat cable.

APPENDIX C

μPB8214-C PRIORITY INTERRUPT CONTROLLER

The μPB8214 is an eight-level priority interrupt controller. Designed to simplify interrupt driven microcomputer systems, the μPB8214 requires a single +5 V power supply and is packed in a 24-pin plastic Dual Inline Package (DIP).

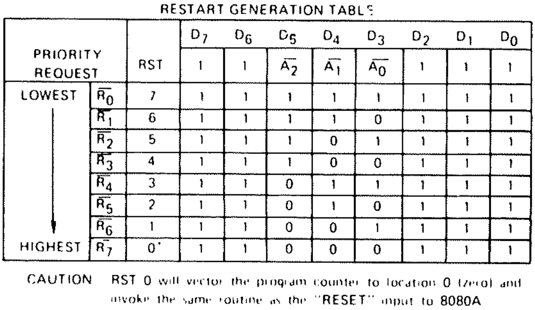
The μPB8214 accepts up to eight interrupts, determines which one has the highest priority, then compares that priority with a software created current status register. If the incoming request has a higher priority than the interrupt currently being serviced, an interrupt request to the processor is generated. Vector information that identifies the interrupting device is also generated.

The interrupt structure of the microcomputer system can be expanded beyond eight interrupt levels by cascading μPB8214s. The μPB8214's interrupt and vector information output is open collector, and control signals are provided to simplify expansion of the interrupt structure. The pin configuration is shown in Figure C-1. The block diagram is illustrated in Figure C-2.

FEATURES

- Eight Priority Levels
- Current Status Register and Priority Comparator
- Easily Expanded Interrupt Structure
- Single +5 Volt Supply

Table C-1 Restart Generation Table



RESTART GENERATION TABLE

PRIORITY REQUEST		RST	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
			1	1	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$	1	1	1
LOWEST	$\overline{R_0}$	7	1	1	1	1	1	1	1	1
↓	$\overline{R_1}$	6	1	1	1	1	0	1	1	1
	$\overline{R_2}$	5	1	1	1	0	1	1	1	1
	$\overline{R_3}$	4	1	1	1	0	0	1	1	1
	$\overline{R_4}$	3	1	1	0	1	1	1	1	1
	$\overline{R_5}$	2	1	1	0	1	0	1	1	1
	$\overline{R_6}$	1	1	1	0	0	1	1	1	1
HIGHEST	$\overline{R_7}$	0*	1	1	0	0	0	1	1	1

CAUTION RST 0 will vector the program counter to location 0 (zero) and invoke the same routine as the "RESET" input to 8080A

Current Status Register

The current status register prevents an incoming interrupt request from overriding the servicing of an interrupt with higher priority. By software, the priority level of the interrupt being serviced by the microprocessor is written into the current status register on $\overline{B}_0$ to $\overline{B}_2$. The bit pattern written should be the complement of the interrupt level.

The interrupt level currently being serviced is written into the current status register by driving $\overline{ECS}$ (Enable Current Status) low. The μPB8214 only accepts interrupts with a higher priority than the value contained by the current status register. Note that the programmer is free to use the current status register for other than as described. Other levels can be written into it. The comparison can be completely disabled by driving $\overline{SGS}$ (Status Group Select) low when $\overline{ECS}$ is driven low. This action causes the μPB8214 to accept incoming interrupts only on the basis of their priority to each other.

Priority Comparator

The priority comparator circuitry compares the level of the interrupt accepted by the priority encoder and request latch with the contents of the current status register. If the incoming request has a priority level higher than that of the current status register, the $\overline{INT}$ output is enabled. Note that this comparison can be disabled by loading the current status register with $\overline{\text{SGS}}$ 0.

Expansion Control Signals

A microcomputer design can often require more than eight interrupts. The μPB8214 is designed so that interrupt system expansion is easily performed by the use of three signals: ETLG (Enable This Level Group); ENLG (Enable Next Level Group); and $\overline{\text{ELR}}$ (Enable Level Read). A high input to ETLG indicates that the μPB8214 can accept an interrupt. In a typical system, the ENLG output from one μPB8214 is connected to the ETLG input of another μPB8214, etc. The ETLG of the μPB8214 with the highest priority is tied high. This configuration sets up priority among the cascaded μPB8214s. The ENLG output will be high for any device that does not have an interrupt pending, thereby allowing a device with lower priority to accept interrupts. The $\overline{\text{ELR}}$ input is basically an IC enable and allows hardware or software to selectively disable/enable individual μPB8214s. A low on the $\overline{\text{ELR}}$ input enables the device.

Interrupt Control Circuitry

The μPB8214 contains two flip-flops and several gates that determine whether an accepted interrupt request to the μPB8214 generates a system interrupt to the 8080A. A condition gate drives the D input of the interrupt flip-flop whenever an interrupt request has been completely accepted. This action requires that the ETLG (Enable This Level Group) and INTE (Interrupt Enable) inputs to the μPB8214 are high, the $\overline{\text{ELR}}$ input is low; the incoming request must be of a higher priority than the contents of the current status register, and the μPB8214 must have been enabled to accept interrupt requests by the clearing of the interrupt disable flip-flop.

Once the condition gate drives the D input of the interrupt flip-flop high, a system interrupt ($\overline{\text{INT}}$) to the 8080A is generated on the next rising edge of the $\overline{\text{CLK}}$ input to the μPB8214. This $\overline{\text{CLK}}$ input is typically connected to the $\phi 2$

(TTL) output of an 8224 so that 8080A set-up time specifications are met. When INT is generated, it sets the interrupt disable flip-flop so that no additional system interrupts are generated until it is reset. It is reset by driving ECS (Enable Current Status) low, thereby writing into the current status register.

It should be noted that the open collector INT from the μPB8214 is active for only one clock period and thus must be externally latched for input to the 8080A. Also, because the INT output is open collector, when μPB8214s are cascaded, an INT output from any one sets all of the interrupt disable flip-flops in the array. Each μPB8214's interrupt disable flip-flop must then be cleared individually to generate subsequent system interrupts. Various parameters and characteristics of the μPB8214 are shown in Figures C-3 through C-5 and listed in Tables C-2 through C-4.

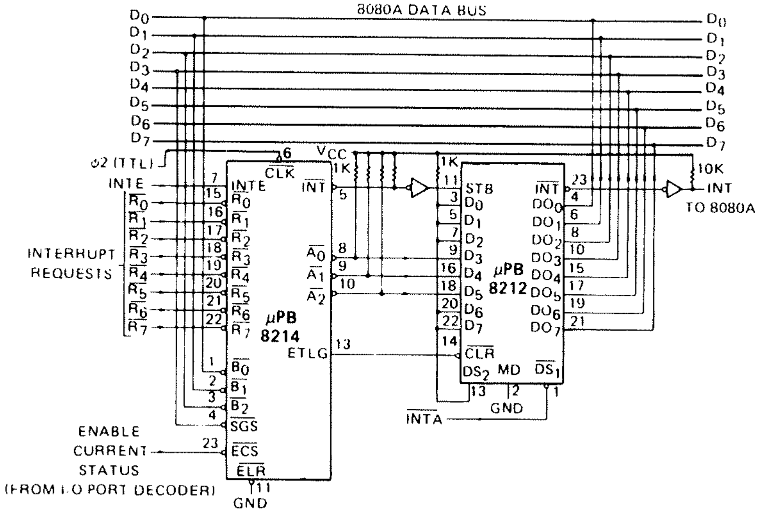
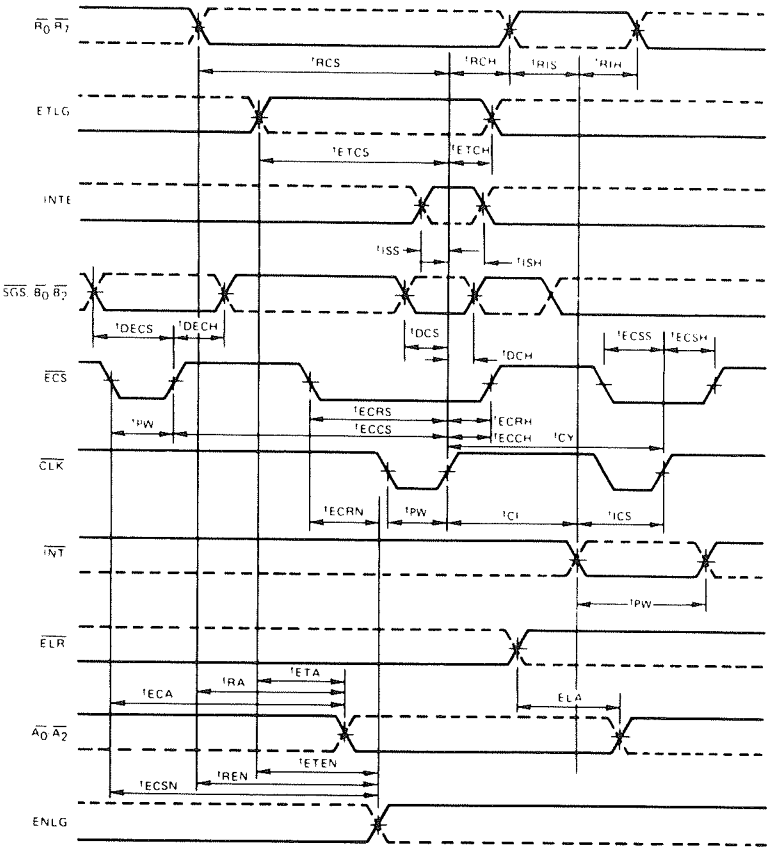
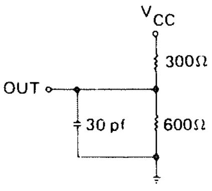


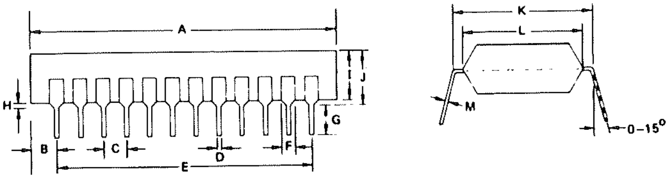
Figure C-3 Typical μPB8214 Circuitry





TEST CIRCUIT

Figure C-4 Timing Waveforms



ITEM	MILLIMETERS	INCHES
A	33 MAX.	1.28
B	2.53	0.1
C	2.54	0.1
D	0.5 ± 0 1	0.02 ± 0.004
E	27.94	1.1
F	1.5	0.059
G	3.2 MIN.	0.125 MIN.
H	0.5 MIN.	0.02 MIN.
I	5.22 MAX.	0.205 MAX.
J	5.72 MAX.	0.225 MAX.
K	15.24	0.6
L	13.2	0.52
M	0.25 ± 0.1	0.01 ± 0.004

Figure C-5 μPB8214C Outline

Table C-2 DC Characteristics

T_a = 0°C to +70°C, V_{CC} = 5V ± 5%

PARAMETER	SYMBOL	LIMITS			UNIT	TEST CONDITIONS
		MIN.	TYP.①	MAX.		
Input Clamp Voltage (all inputs)	V_C			1.0	V	I_C= 5mA
Input Forward Current: ETLG input	I_F		.15	0.5	mA	V_F=0.45V
all other inputs			.08	0.25	mA	
Input Reverse Current: ETLG input	I_R			80	μA	V_R=5.25V
all other inputs				40	μA	
Input LOW Voltage: all inputs	V_{IL}			0.8	V	V_{CC}=5.0V
Input HIGH Voltage: all inputs	V_{IH}	2.0			V	V_{CC}=5.0V
Power Supply Current	I_{CC}		90	130	mA	②
Output LOW Voltage: all outputs	V_{OL}		.3	.45	V	I_{OL}=10mA
Output HIGH Voltage: ENLG output	V_{OH}	2.4	3.0		V	I_{OH}= 1mA
Short Circuit Output Current: ENLG output	I_{OS}	20	35	55	mA	V_{OS}=0V, V_{CC}=5.0V
Output Leakage Current: $\overline{INT}$ and $\overline{A_0}$–$\overline{A_2}$	I_{CEX}			100	μA	V_{CEX}=5.25V

Table C-3 Capacitance

T_a = 25°C

PARAMETER	SYMBOL	LIMITS			UNIT	TEST CONDITIONS
		MIN.	TYP.①	MAX.		
Input Capacitance	C_{IN}		5	10	pF	V_{BIAS}=2.5V
Output Capacitance	C_{OUT}		7	12	pF	V_{CC}=5V f=1mHz

Table C-4 AC Characteristics

T_a = 0 C to +70 C, V_{CC} = +5V ± 5%

PARAMETER	SYMBOL	LIMITS			UNIT	TEST CONDITIONS
		MIN.	TYP.①	MAX.		
$\overline{CLK}$ Cycle Time	t_{CY}	80	50		ns	Input pulse
$\overline{CLK}$, $\overline{ECS}$, $\overline{INT}$ Pulse Width	t_{PW}	25	15		ns	amplitude 2.5 Volts.
INTE Setup Time to $\overline{CLK}$	t_{ISS}	16	12		ns	
INTE Hold Time after $\overline{CLK}$	t_{ISH}	20	10		ns	
ETLG Setup Time to $\overline{CLK}$	t_{ETCS}④	25	12		ns	Input rise and fall
ETLG Hold Time After $\overline{CLK}$	t_{ETCH}④	20	10		ns	times 5 ns between
$\overline{ECS}$ Setup Time to $\overline{CLK}$	t_{ECCS}④	80	50		ns	1 and 2 Volts
$\overline{ECS}$ Hold Time After $\overline{CLK}$	t_{ECCH}⑤	0			ns	
$\overline{ECS}$ Setup Time to $\overline{CLK}$	t_{ECRS}⑤	110	70		ns	
$\overline{ECS}$ Hold Time After $\overline{CLK}$	t_{ECRH}⑤	0				Output loading of
$\overline{ECS}$ Setup Time to $\overline{CLK}$	t_{ECSS}④	75	70		ns	15 mA and 30 pF
$\overline{ECS}$ Hold Time After $\overline{CLK}$	t_{ECSH}①	0			ns	
$\overline{SGS}$ and $\overline{B_0}-\overline{B_2}$ Setup Time to $\overline{CLK}$	t_{DCS}④	70	50		ns	
$\overline{SGS}$ and $\overline{B_0}-\overline{B_2}$ Hold Time After $\overline{CLK}$	t_{DCH}④	0			ns	Speed measurements
$\overline{R_0}$ $\overline{R_7}$ Setup Time to $\overline{CLK}$	t_{RCS}⑤	90	55		ns	taken at the 1.5 Volts
$\overline{R_0}$- $\overline{R_7}$ Hold Time After $\overline{CLK}$	t_{RCH}⑤	0			ns	levels
$\overline{INT}$ Setup Time to $\overline{CLK}$	t_{ICS}	55	35		ns	
$\overline{CLK}$ to $\overline{INT}$ Propagation Delay	t_{CI}		15	25	ns	
$\overline{R_0}$ - $\overline{R_7}$ Setup Time to $\overline{INT}$	t_{RIS}⑥	10	0		ns	
$\overline{R_0}$ $\overline{R_7}$ Hold Time After $\overline{INT}$	t_{RIH}⑥	35	20		ns	
$\overline{R_0}$ - $\overline{R_7}$ to $\overline{A_0}$ - $\overline{A_2}$ Propagation Delay	t_{RA}		80	100	ns	
$\overline{ELR}$ to $\overline{A_0}$ $\overline{A_2}$ Propagation Delay	t_{ELA}		40	55	ns	
$\overline{ECS}$ to $\overline{A_0}$ $\overline{A_2}$ Propagation Delay	t_{ECA}		100	120	ns	
ETLG to $\overline{A_0}$- $\overline{A_2}$ Propagation Delay	t_{ETA}		35	70	ns	
$\overline{SGS}$ and $\overline{B_0}$ - $\overline{B_2}$ Setup Time to $\overline{ECS}$	t_{DECS}⑥	15	10		ns	
$\overline{SGS}$ and $\overline{B_0}$ - $\overline{B_2}$ Hold Time After $\overline{ECS}$	t_{DECH}⑥	15	10		ns	
$\overline{R_0}$ $\overline{R_7}$ to ENLG Propagation Delay	t_{REN}		45	70	ns	
ELTG to ENLG Propagation Delay	t_{ETEN}		20	25	ns	
$\overline{ECS}$ to ENLG Propagation Delay	t_{ECRN}		85	90	ns	
$\overline{ECS}$ to ENLG Propagation Delay	t_{ECSN}		35	55	ns	

1 Typical values are for T_a = 25°C, V_{CC} = 5.0V

2 B_0–B_2, SGS, CLK, R_0–R_4 grounded, all other inputs and all outputs open.

3 This parameter is periodically sampled and not 100% tested.

4 Required for proper operation if NTE is enabled during next clock pulse.

5 These times are not required for proper operation but for desired change in interrupt flip-flop.

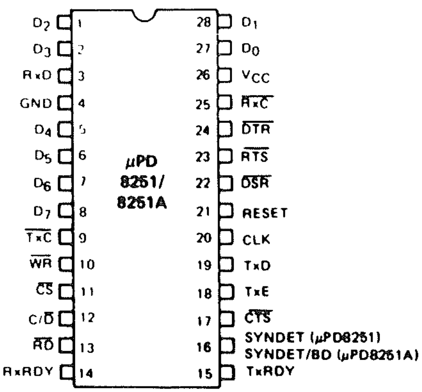
6 Required for new request or status to be properly loaded.

APPENDIX D

μPD8251-C PROGRAMMABLE COMMUNICATION INTERFACES

The μPD8251 and μPD8251A Universal Synchronous Asynchronous receiver/Transmitters (USARTs) (see Figure D-1) are designed for microcomputer systems data communications. The USART is used as a peripheral and is programmed by the μPD8080 or other processor to communicate in commonly used serial data transmission techniques including IBM Bi-Sync. The USART receives serial data streams and converts them into parallel data characters for the processor. While receiving serial data, the USART also accepts data characters from the processor in parallel format, converts them to serial format and transmits. The USART signals the processor when it has completely received or transmitted a character and requires service. Complete USART status, including data format errors and control signals such as TxE and SYNDET, is available to the processor at any time.

PIN CONFIGURATION



PIN NAMES

D_7-D_0	Data Bus (8 bits)
C/$\overline{D}$	Control or Data is to be Written or Read
$\overline{RD}$	Read Data Command
$\overline{WR}$	Write Data or Control Command
$\overline{CS}$	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
$\overline{TxC}$	Transmitter Clock (TTL)
TxD	Transmitter Data
$\overline{RxC}$	Receiver Clock (TTL)
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080)
TxRDY	Transmitter Ready (ready for char. from 8080)
$\overline{DSR}$	Data Set Ready
$\overline{DTR}$	Data Terminal Ready
SYNDET	Sync Detect
SYNDET/BD	Sync Detect/Break Detect
$\overline{RTS}$	Request to Send Data
$\overline{CTS}$	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

Figure D-1 μPD8251/8251A Pin Configuration

FEATURES

- Asynchronous or synchronous operation
  - Asynchronous:

    5 to 8-bit characters

    Clock rate — 1, 16 or 64 × baud rate

    Break character generation

    Select 1, 1-1/2 or 2 stop bits

    False start bit detector

    Automatic break detect and handling (μPD8251A)
  - Synchronous:

    5 to 8-bit characters

    Internal or external character synchronization

    Automatic sync insertion

    Single or double sync characters
- Baud Rate (1X Model) — dc to 56K baud (μPD8251)
  — dc to 64K baud (μPD8251A)
- Full duplex, double buffered transmitter and receiver
- Parity, overrun, and framing flags
- Fully compatible with 8080/8085/μPD780 (Z80^{TM})
- All inputs and outputs are TTL-compatible
- Single +5 Volt supply
- Separate device receive and transmit TTL clocks
- 28-pin plastic DIP package
- N-channel MOS technology

TM: Z80 is a registered trademark of Zilog.

FUNCTIONAL DESCRIPTION

The μPD8251 and μPD8251A USARTs are designed specifically for 8080 microsystems but work with most 8-bit processors. Operation of the μPD8251 or μPD8251A, like other I/O devices in the 8080 family, are programmed by system software for maximum flexibility.

In the receive mode, the μPD8251 or μPD8251A converts incoming serial format data into parallel data and makes certain format checks. In the transmit mode, it formats parallel data into serial form. The device also supplies or removes characters or bits that are unique to the communication format in use. By performing conversion and formatting services automatically, the USART appears to the processor as a simple or "transparent" input or output of byte-oriented parallel data.

μPD8251A FEATURES AND ENHANCEMENTS

The μPD8251A is an advanced design of the industry standard 8251 USART. It operates with a wide range of microprocessors, including the 8080, 8085, and μPD780 (Z80^{TM}). The additional features and enhancements of the μPD8251A over the μPD8251 are listed below. The basic operation is shown in Table D-1. A block diagram of the μPD8251A is shown in Figure D-2.

1. The data paths are double-buffered with separate I/O registers for control, status, Data In and Data Out. This feature simplifies control programming and minimizes processor overhead.
2. The Receiver detects and handles "break" automatically in asynchronous operations, which relieves the processor of this task.
3. The Receiver is prevented from starting when in "break" state by a refined Rx initialization. This also prevents a disconnected USART from causing unwanted interrupts.
4. When a transmission is concluded, the TxD line returns to the marking state unless SBRK is programmed.

5. The Tx Disable command is prevented from halting transmission by the Tx Enable Logic enhancement until all data previously written has been transmitted. The same logic also prevents the transmitter from turning off in the middle of a word.
6. Internal Sync Detect is disabled when External Sync Detect is programmed. An External Detect Status is provided through a flip-flop that clears itself on a status read.
7. The possibility of a false sync detect is minimized by:
   - ensuring that if a double sync character is programmed, the characters be contiguously detected.
   - clearing the Rx register to all Logic 1s (V_{OH}) whenever the Enter Hunt command is issued in Sync mode.
8. The $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device as long as the µPD8251A is not selected.
9. The µPD8251A Status can be read at any time; however, the status update is inhibited during status read.
10. The µPD8251A has enhanced AC and DC characteristics and is free from extraneous glitches, providing higher speed and improved operating margins.
11. Baud rate from DC to 64K.

Table D-1 Basic Operation

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	µPD8251/µPD8251A → Data Bus
0	1	0	0	Data Bus → µPD8251/µPD8251A
1	0	1	0	Status → Data Bus
1	1	0	0	Data Bus → Control
X	X	X	1	Data Bus → 3-State
X	1	1	0	

TM:Z80 is a registered trademark of Zilog.

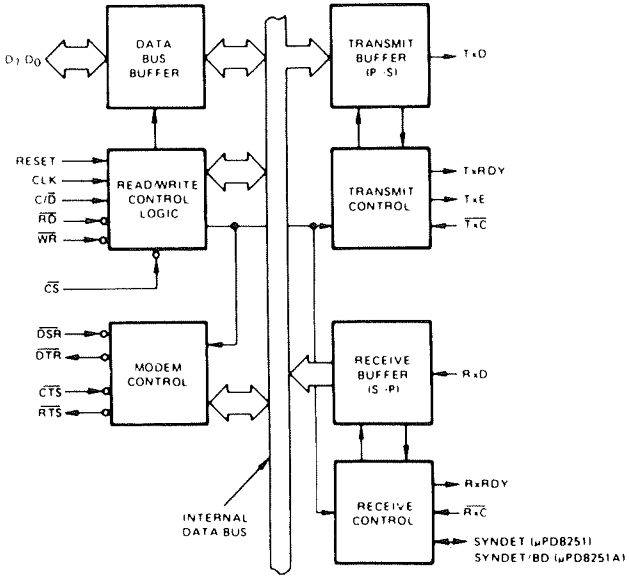


Figure D-2 Block Diagram

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature	−0°C to +70°C
Storage Temperature	−65°C to +125°C
All Output Voltages	−0.5 to +7 Volts
All Input Voltages	−0.5 to +7 Volts
Supply Voltages	−0.5 to +7 Volts

COMMENT: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*$T_a = 25°C$

DC characteristics, capacitance, and AC characteristics are listed in Tables D-2 through D-4.

Table D-2 DC Characteristics

T_a = 0°C to 70°C; V_{CC} = 5.0V ± 5%; GND = 0V

PARAMETER	SYMBOL	LIMITS					UNIT	TEST CONDITIONS
		µPD8251			µPD8251A			
		MIN	TYP	MAX	MIN	MAX		
Input Low Voltage	V_{IL}	−0.5		0.8	0.5	0.8	V	
Input High Voltage	V_{IH}	2.0		V_{CC}	2.0	V_{CC}	V	
Output Low Voltage	V_{OL}			0.45		0.45	V	µPD8251 I_{OL} = 1.7 mA µPD8251A I_{OL} = 2.2 mA
Output High Voltage	V_{OH}	2.4			2.4		V	µPD8251 I_{OH} = −100 µA µPD8251A I_{OH} = −400 µA
Data Bus Leakage	I_{DL}			−50		−10	µA	V_{OUT} = 0.45V
				10		10		V_{OUT} = V_{CC}
Input Load Current	I_{IL}			10		10	µA	At 5.5V
Power Supply Current	I_{CC}		45	80		100	mA	µPD8251A All Outputs = Logic 1

Table D-3 Capacitance

T_a = 25°C, V_{CC} = GND = 0V

PARAMETER	SYMBOL	LIMITS			UNIT	TEST CONDITIONS
		MIN	TYP	MAX		
Input Capacitance	C_{IN}			10	pF	fc = 1 MHz
I/O Capacitance	$C_{I/O}$			20	pF	Unmeasured pins returned to GND

Table D-4 AC Characteristics

T_a 0°C to 70°C V_{CC} 5.0V ± 5% GND = 0V

PARAMETER	SYMBOL	LIMITS				UNIT	TEST CONDITIONS
		µPD8251		µPD8215A			
		MIN	MAX	MIN	MAX		
READ							
Address Stable before $\overline{READ}$, ($\overline{CS}$, $C/\overline{D}$)	t_{AR}	50		0		ns	
Address Hold Time for $\overline{READ}$, ($\overline{CS}$, $C/\overline{D}$)	t_{RA}	5		0		ns	
$\overline{READ}$ Pulse Width	t_{RR}	430		250		ns	
Data Delay from $\overline{READ}$	t_{RD}		350		200	ns	µPD8251 C_L = 100 pF µPD8251A C_L = 150 pF
$\overline{READ}$ to Data Floating	t_{DF}	25	200	10	100	ns	µPD8251 C_L = 100 pF C_L = 15 pF
WRITE							
Address Stable before $\overline{WRITE}$	t_{AW}	20		0		ns	
Address Hold Time for $\overline{WRITE}$	t_{WA}	20		0		ns	
$\overline{WRITE}$ Pulse Width	t_{WW}	400		250		ns	
Data Set Up Time for $\overline{WRITE}$	t_{DW}	200		150		ns	
Data Hold Time for WRITE	t_{WD}	40		0		ns	
Recovery Time Between WRITES ②	t_{RV}	6		6		t_{CY}	
OTHER TIMING							
Clock Period ③	t_{CY}	0.420	1.35	0.32	1.35	µs	
Clock Pulse Width High	$t_{\phi W}$	220	0.7t_{CY}	120	t_{CY} 90	ns	
Clock Pulse Width Low	$t_{\phi W}$			90		ns	
Clock Rise and Fall Time	t_R, t_F	0	50	5	20	ns	
TxD Delay from Falling Edge of TxC	t_{DTx}		1		1	µs	
Rx Data Set Up Time to Sampling Pulse	t_{SRx}	2		2		µs	µPD8251 C_L = 100 pF
Rx Data Hold Time to Sampling Pulse	t_{HRx}	2		2		µs	
Transmitter Input Clock Frequency	f_{Tx}						
1X Baud Rate		DC	56		64	kHz	
16X Baud Rate		DC	520		310	kHz	
64X Baud Rate		DC	520		615	kHz	
Transmitter Input Clock Pulse Width	t_{TPW}						
1X Baud Rate		12		12		t_{CY}	
16X and 64X Baud Rate		1		1		t_{CY}	
Transmitter Input Clock Pulse Delay	t_{TPD}						
1X Baud Rate		15		15		t_{CY}	
16X and 64X Baud Rate		3		3		t_{CY}	
Receiver Input Clock Frequency	f_{Rx}						
1X Baud Rate		DC	56		64	kHz	
16X Baud Rate		DC	520		310	kHz	
64X Baud Rate		DC	520		615	kHz	
Receiver Input Clock Pulse Width	t_{RPW}						
1X Baud Rate		12		12		t_{CY}	
16X and 64X Baud Rate		1		1		t_{CY}	
Receiver Input Clock Pulse Delay	t_{RPD}						
1X Baud Rate		15		15		t_{CY}	
16X and 64X Baud Rate		3		3		t_{CY}	
TxRDY Delay from Center of Data Bit	t_{Tx}		16		8	t_{CY}	µPD8251 C_L = 50 pF
RxRDY Delay from Center of Data Bit	t_{RX}		20		24	t_{CY}	
Internal SYNDET Delay from Center of Data Bit	t_{IS}		25		24	t_{CY}	
External SYNDET Set Up Time before Falling Edge of $\overline{RxC}$	t_{ES}		16		16	t_{CY}	
TxEMPTY Delay from Center of Data Bit	t_{TxE}		16		20	t_{CY}	µPD8251 C_L = 50 pF
Control Delay from Rising Edge of WRITE (TxE, $\overline{DTR}$, $\overline{RTS}$)	t_{WC}		16		8	t_{CY}	
Control to READ Set Up Time ($\overline{DSR}$, $\overline{CTS}$)	t_{CR}		16		20	t_{CY}	

Notes: ① AC timings measured at V_{OH} = 2.0 V_{OL} = 0.8, and with load circuit of Figure 1

② This recovery time is for initialization only, when MODE, SYNC1, SYNC2, COMMAND and first DATA BYTES are written into the USART. Subsequent writing of both COMMAND and DATA are only allowed when TxRDY = 1

③ The TxC and RxC frequencies have the following limitations with respect to CLK

For 1X Baud Rate, f_{Tx} or f_{Rx} ≤ 1/(30 t_{CY})

For 16X and 64X Baud Rate f_{TX} or f_{Rx} ≤ 1/(4.5 t_{CY})

④ Reset Pulse Width = 6 t_{CY} minimum

A test load circuit is shown in Figure D-3.

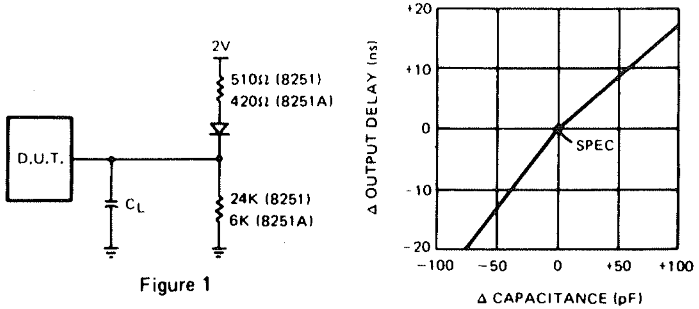


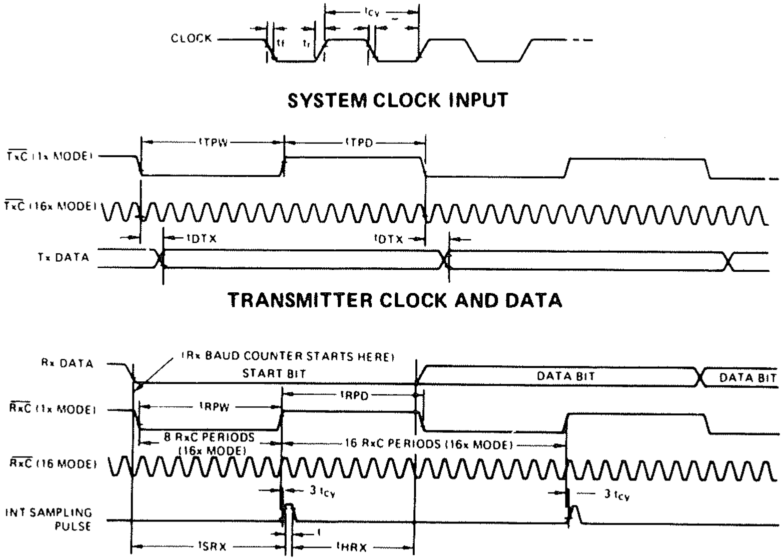
Figure 1

Typical Δ Output Delay Versus Δ Capacitance (pF)

TEST LOAD CIRCUIT

Figure D-3 Test Load Circuit

Various timing waveforms are shown in Figure D-4.



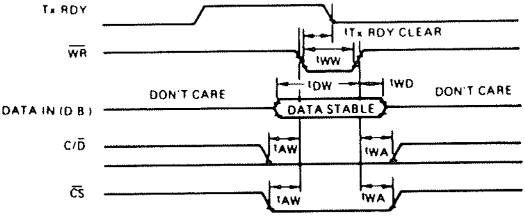
SYSTEM CLOCK INPUT

TRANSMITTER CLOCK AND DATA

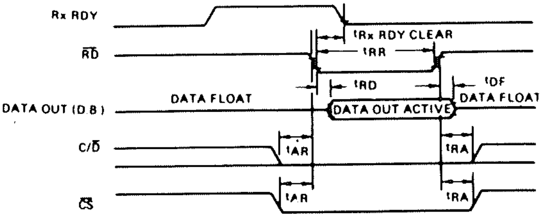
RECEIVER CLOCK AND DATA

Figure D-4 Timing Waveforms

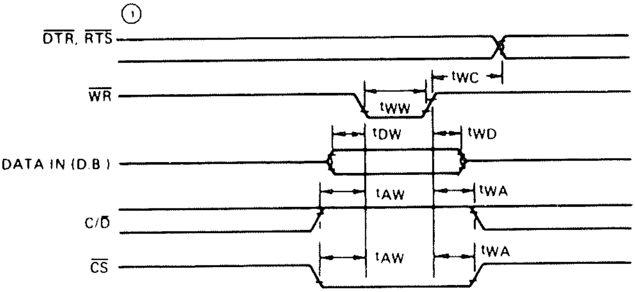
TIMING WAVEFORM
(CONT.)



WRITE DATA CYCLE (PROCESSOR → USART)

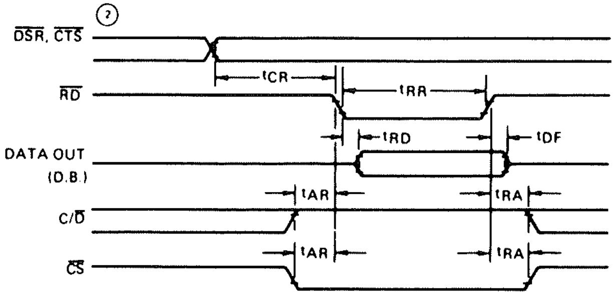


READ DATA CYCLE (PROCESSOR ← USART)



**WRITE CONTROL OR OUTPUT PORT CYCLE
(PROCESSOR → USART)**

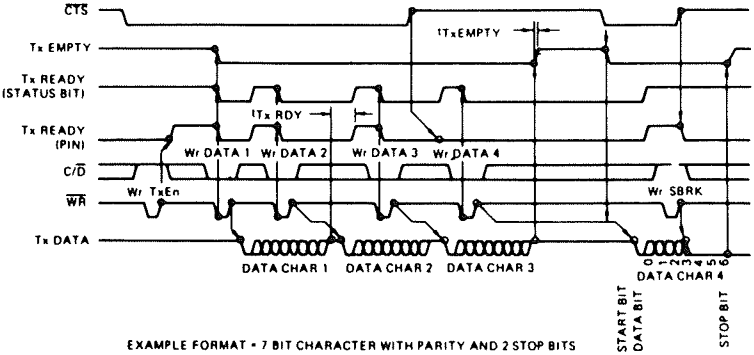
Figure D-4 Timing Waveforms (cont'd)



READ CONTROL OR INPUT PORT CYCLE (PROCESSOR ← USART)

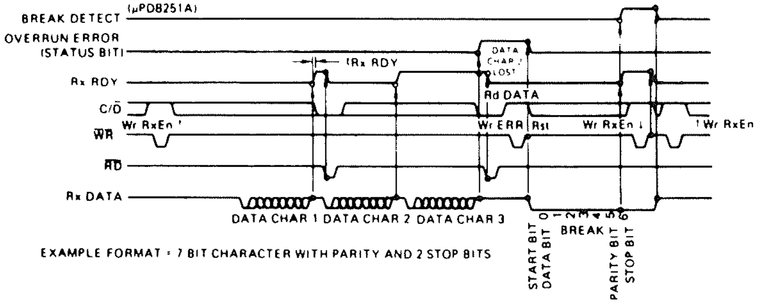
NOTES ① T_{WC} Includes the response timing of a control byte.

② T_{CR} Includes the effect of CTS on the TxENBL circuitry



EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY AND 2 STOP BITS

TRANSMITTER CONTROL AND FLAG TIMING (ASYNC MODE)

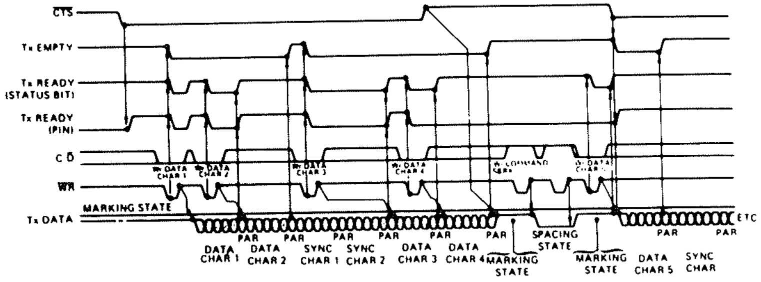


EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY AND 2 STOP BITS

RECEIVER CONTROL AND FLAG TIMING (ASYNC MODE)

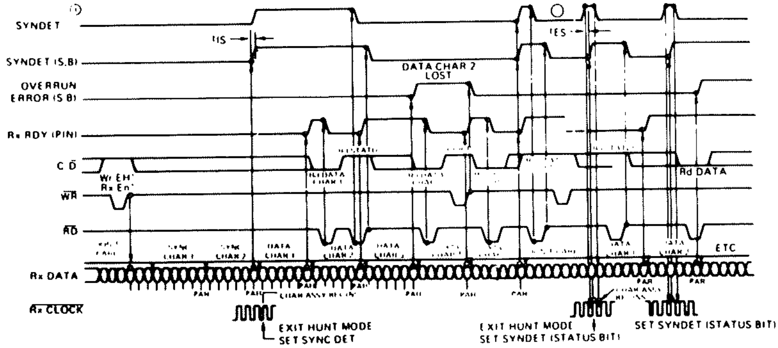
Figure D-4 Timing Waveforms (cont'd)

TIMING WAVEFORM (CONT.)



EXAMPLE FORMAT • 5 BIT CHARACTER WITH PARITY AND 2 SYNC CHARACTERS

TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)



RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)

Notes: ① Internal sync, 2 sync characters, 5 bits, with parity
② External sync, 5 bits, with parity

Figure D-4 Timing Waveforms (cont'd)

Pin identification for the USART is given in Table D-5.

Table D-5 Pin Identification

PIN			FUNCTION
NO.	SYMBOL	NAME	
1, 2, 27, 28 5 - 8	$D_7 - D_0$	Data Bus Buffer	An 8-bit, 3-state bi-directional buffer used to interface the USART to the processor data bus. Data is transmitted or received by the buffer in response to input/output or Read Write instructions from the processor. The Data Bus Buffer also transfers Control words Command words, and Status
26	V_{CC}	V_{CC} Supply Voltage	+5 volt supply
4	GND	Ground	Ground
Read Write Control Logic			This logic block accepts inputs from the processor Control Bus and generates control signals for overall USART operation. The Mode Instruction and Command Instruction registers that store the control formats for device functional definition are located in the Read Write Control Logic
21	RESET	Reset	A "one" on this input forces the USART into the "Idle" mode where it will remain until reinitialized with a new set of control words. Minimum RESET pulse width is 6 t_{CY}
20	CLK	Clock Pulse	The CLK input provides for internal device timing and is usually connected to the Phase 2 (TTL) output of the μPB8224 Clock Generator. External inputs and outputs are not referenced to CLK but the CLK frequency must be at least 30 times the Receiver or Transmitter clocks in the synchronous mode and 4 5 times for the asynchronous mode.
10	$\overline{WR}$	Write Data	A "zero" on this input instructs the USART to accept the data or control word which the processor is writing out on the data bus
13	$\overline{RD}$	Read Data	A "zero" on this input instructs the USART to place the data or status information onto the Data Bus for the processor to read
12	$C\ \overline{D}$	Control Data	The Control Data input, in conjunction with the WR and $\overline{RD}$ inputs, informs the USART to accept or provide either a data character, control word or status information via the Data Bus 0 Data 1 Control
11	$\overline{CS}$	Chip Select	A "zero" on this input enables the USART to read from or write to the processor
Modem Control			The μPD8251 and μPD8251A have a set of control inputs and outputs which may be used to simplify the interface to a Modem
22	$\overline{DSR}$	Data Set Ready	The Data Set Ready input can be tested by the processor via Status information. The $\overline{DSR}$ input is normally used to test Modem Data Set Ready condition
24	$\overline{DTR}$	Data Terminal Ready	The Data Terminal Ready output can be controlled via the Command word. The $\overline{DTR}$ output is normally used to drive Modem Data Terminal Ready or Rate Select lines
23	$\overline{RTS}$	Request to Send	The Request to Send output can be controlled via the Command word. The $\overline{RTS}$ output is normally used to drive the Modem Request to Send line
17	$\overline{CTS}$	Clear to Send	A "zero" on the Clear to Send input enables the USART to transmit serial data if the TxEN bit in the Command Instruction register is enabled (one)

TRANSMIT BUFFER

The Transmit Buffer receives parallel data from the Data Bus Buffer on the internal data bus, converts parallel to serial data, inserts the necessary characters or bits needed for the programmed communication format and sends composite serial data on the TxD pin.

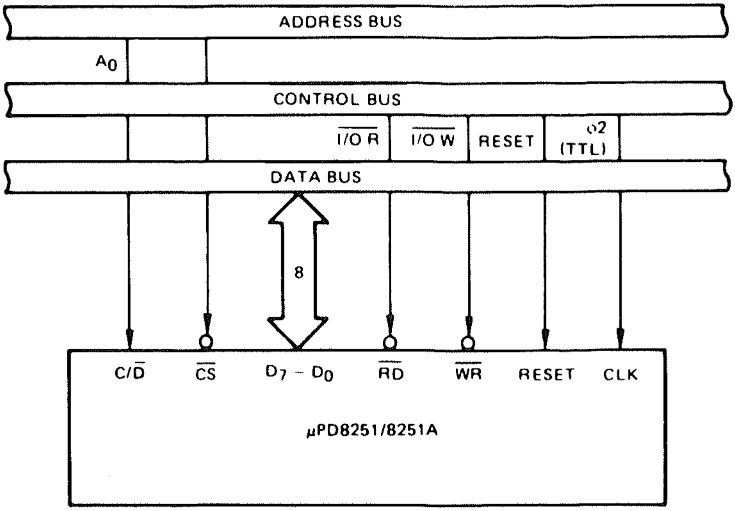
Pin identification for the transmit buffer of the USART is given in Table D-6.

Table D-6 Pin Identification

PIN			FUNCTION
NO.	SYMBOL	NAME	
Transmit Control Logic			The Transmit Control Logic accepts and outputs all external and internal signals necessary for serial data transmission.
15	TxRDY	Transmitter Ready	Transmitter Ready signals the processor that the transmitter is ready to accept a data character. TxRDY can be used as an interrupt or may be tested through the Status information for polled operation. Loading a character from the processor automatically resets TxRDY, on the leading edge.
18	TxE	Transmitter Empty	The Transmitter Empty output signals the processor that the USART has no further characters to transmit. TxE is automatically reset upon receiving a data character from the processor. In half-duplex, TxE can be used to signal end of a transmission and request the processor to "turn the line around." The TxEn bit in the command instruction does not effect TxE. In the Synchronous mode, a "one" on this output indicates that a Sync character or characters are about to be automatically transmitted as "fillers" because the next data character has not been loaded.
9	$\overline{\text{TxC}}$	Transmitter Clock	The Transmitter Clock controls the serial character transmission rate. In the Asynchronous mode, the $\overline{\text{TxC}}$ frequency is a multiple of the actual Baud Rate. Two bits of the Mode Instruction select the multiple to be 1x, 16x, or 64x the Baud Rate. In the Synchronous mode, the $\overline{\text{TxC}}$ frequency is automatically selected to equal the actual Baud Rate. Note that for both Synchronous and Asynchronous modes, serial data is shifted out of the USART by the falling edge of $\overline{\text{TxC}}$.
19	TxD	Transmitter Data	The Transmit Control Logic outputs the composite serial data stream on this pin.

The USART interface to the standard system bus is shown in Figure D-5.

Figure D-5 μPD8251 and μPD8251A Interface to 8080 Standard System Bus



RECEIVE BUFFER

The Receive Buffer accepts serial data input at the $\overline{\text{RxD}}$ pin and converts the data from serial to parallel format. Bits or characters required for the specific communication technique in use are checked and then an eight-bit "assembled" character is readied for the processor. For communication techniques that require less than eight bits, the μPD8251 and μPD8251A set the extra bits to "zero".

The pin identification for the Receive buffer of the USART is given in Table D-7.

Table D-7 Pin Identification

PIN			FUNCTION
NO	SYMBOL	NAME	
Receiver Control Logic			This block manages all activities related to incoming data.
14	RxRDY	Receiver Ready	The Receiver Ready output indicates that the Receiver Buffer is ready with an "assembled" character for input to the processor. For Polled operation, the processor can check RxRDY using a Status Read or RxRDY can be connected to the processor interrupt structure. Note that reading the character to the processor automatically resets RxRDY.
25	$\overline{\text{RxC}}$	Receiver Clock	The Receiver Clock determines the rate at which the incoming character is received. In the Asynchronous mode, the $\overline{\text{RxC}}$ frequency may be 1.16 or 64 times the actual Baud Rate but in the Synchronous mode the $\overline{\text{RxC}}$ frequency must equal the Baud Rate. Two bits in the mode instruction select Asynchronous at 1x, 16x or 64x or Synchronous operation at 1x the Baud Rate. Unlike $\overline{\text{TxC}}$, data is sampled by the μPD8251 and μPD8251A on the rising edge of $\overline{\text{RxC}}$. ①
3	RxD	Receiver Data	A composite serial data stream is received by the Receiver Control Logic on this pin
16	SYNDET (μPD8251)	Sync Detect	The SYNC Detect pin is only used in the Synchronous mode. The μPD8251 may be programmed through the Mode Instruction to operate in either the internal or external Sync mode and SYNDET then functions as an output or input respectively. In the internal Sync mode, the SYNDET output will go to a "one" when the μPD8251 has located the SYNC character in the Receive mode. If double SYNC character (bi-sync) operation has been programmed, SYNDET will go to "one" in the middle of the last bit of the second SYNC character. SYNDET is automatically reset to "zero" upon a Status Read or RESET. In the external SYNC mode, a "zero" to "one" transition on the SYNDET input will cause the μPD8251 to start assembling data character on the next falling edge of $\overline{\text{RxC}}$. The length of the SYNDET input should be at least one $\overline{\text{RxC}}$ period, but may be removed once the μPD8251 is in SYNC

Table D-7 Pin Identification (cont.)

PIN			FUNCTION
NO	SYMBOL	NAME	
16	SYNDET/BD (μPD8251A)	Sync Detect/ Break Detect	The SYNDET/BD pin is used in both Synchronous and Asynchronous modes. When in SYNC mode the features for the SYNDET pin described above apply. When in Asynchronous mode, the Break Detect output will go high when an all zero word of the programmed length is received. This word consists of: start bit, data bit, parity bit and one stop bit. Reset only occurs when Rx data returns to a logic one state or upon chip reset. The state of Break Detect can be read as a status bit.

Note: ① Since the μPD8251 and μPD8251A will frequently be handling both the reception and transmission for a given link, the Receive and Transmit Baud Rates will be same. $\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ then require the same frequency and may be tied together and connected to a single clock source or Baud Rate Generator.

Examples: If the Baud Rate equals 110 (Async):

$\overline{\text{RxC}}$ or $\overline{\text{TxC}}$ equals 110 Hz (1x)
$\overline{\text{RxC}}$ or $\overline{\text{TxC}}$ equals 1.76 KHz (16x)
$\overline{\text{RxC}}$ or $\overline{\text{TxC}}$ equals 7.04 KHz (64x)

If the Baud Rate equals 300:

$\overline{\text{RxC}}$ or $\overline{\text{TxC}}$ equals 300 Hz (1x) A or S
$\overline{\text{RxC}}$ or $\overline{\text{TxC}}$ equals 4800 Hz (16x) A only
$\overline{\text{RxC}}$ or $\overline{\text{TxC}}$ equals 19.2 KHz (64x) A only

OPERATIONAL DESCRIPTION

A set of control words must be sent to the μPD8251 and μPD8251A to define the desired mode and communications format. The control words will specify the BAUD RATE FACTOR (1x, 16x, 64x), CHARACTER LENGTH (5 to 8), NUMBER OF STOP BITS (1, 1-1/2, 2) ASYNCHRONOUS or SYNCHRONOUS MODE, SYNDET (IN or OUT), PARITY, etc.

After receiving the control words, the μPD8251 and μPD8251A are ready to communicate. TxRDY is raised to signal the processor that the USART is ready to receive a character for transmission. When the processor writes a character to the USART, TxRDY is automatically reset.

Concurrently, the μPD8251 and μPD8251A may receive serial data; and after receiving an entire character, the RxRDY output is raised to indicate a completed character is ready for the processor. The processor fetch will automatically reset RxRDY.

NOTE

The μPD8251 and μPD8251A may provide faulty RxRDY for the first read after power-on or for the first read after receive is re-enabled by a command instruction (RxE). A dummy read is recommended to clear faulty RxRDY. But this is not the case for the first read after hardware or software reset after the device operation has once been established.

The μPD8251 and μ PD8251A cannot transmit until the TxEN (Transmitter Enable) bit has been set by a Command Instruction and until the $\overline{\text{CTS}}$ (Clear to Send) input is a "zero". TxD is held in the "marking" state after Reset awaiting new control words.

PROGRAMMING FOR USART

The USART must be loaded with a group of two to four control words provided by the processor before data reception and transmission can begin. A Reset (internal or external) must immediately precede the control words which are used to program the complete operational description of the communications interface. If an external RESET is not available, three successive 00 Hex or two successive 80 Hex command instructions (C/$\overline{D}$ = 1) followed by a software reset command instruction (40 Hex) can be used to initialize the µPD8251 and µPD8251A.

There are two control word formats:

1. Mode Instruction
2. Command Instruction

MODE INSTRUCTION

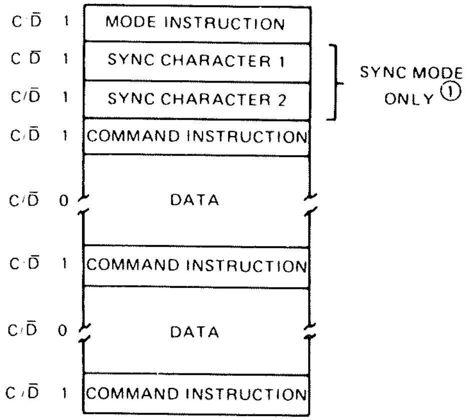
This control word specifies the general characteristics of the interface regarding the SYNCHRONOUS or ASYNCHRONOUS MODE, BAUD RATE FACTOR, CHARACTER LENGTH, PARITY, and NUMBER OF STOP BITS. Once the Mode Instruction has been received, SYNC characters or Command Instructions may be inserted depending on the Mode Instruction content.

COMMAND INSTRUCTION

This control word will be interpreted as a SYNC character definition if immediately preceded by a Mode Instruction that specified a synchronous format. After the SYNC character(s) are specified or after an Asynchronous Mode Instruction, all subsequent control words will be interpreted as an update to the Command Instruction. Command Instruction updates may occur at any time during the data block. To modify the Mode Instruction, a bit may be set in the Command Instruction that causes an internal Reset, which allows a new Mode Instruction to be accepted.

TYPICAL DATA BLOCK

A typical data block is shown in Figure D-6.



NOTE ① The second SYNC character is skipped if MODE instruction has programmed the μPD8251 and μPD8251A to single character Internal SYNC Mode. Both SYNC characters are skipped if MODE instruction has programmed the μPD8251 and μPD8251A to ASYNC mode

Figure D-6 Typical Data Block

MODE INSTRUCTION DEFINITION

The μPD8251 and μPD8251A can operate in either Asynchronous or Synchronous communication modes. Understanding how the Mode Instruction controls the functional operation of the USART is easiest when the device is considered to be two separate components, one asynchronous and the other synchronous, which share the same support circuits and PCB. Although the format definition can be changed at will or "on the fly", the two modes are explained separately for clarity.

ASYNCHRONOUS TRANSMISSION

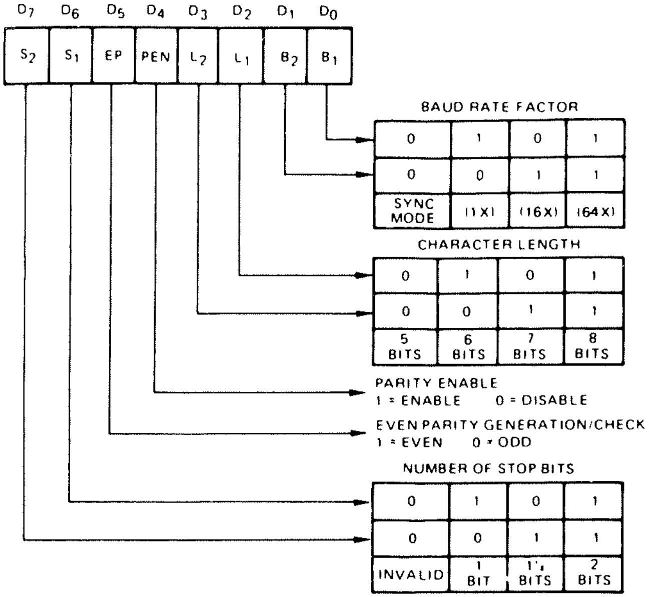
When a data character is written into the μPD8251 and μPD8251A, the USART automatically adds a START bit (low level or "space") and the number of STOP bits (high level or "mark") specified by the Mode Instruction. If Parity has been enabled, an odd or even Parity bit is inserted just before the STOP bit(s), as specified by the Mode Instruction. Then, depending on $\overline{\mathrm{CTS}}$ and TxEN, the character may be transmitted as a serial data stream at the TxD output. Data is shifted out by the falling edge of $\overline{\mathrm{TxC}}$ at $\overline{\mathrm{TxC}}$, $\overline{\mathrm{TxC}}$/16 or $\overline{\mathrm{TxC}}$/64, as defined by the Mode instruction.

If no data characters have been loaded into the μPD8251 and μPD8251A, or if all available characters have been transmitted, the TxD output remains "high" (marking) in preparation for sending the START bit of the next character provided by the processor. TxD may be forced to send a BREAK (continuously low) by setting the correct bit in the Command Instruction.

ASYNCHRONOUS RECEIVE

The RxD input line is normally held "high" (marking) by the transmitting device. A falling edge at RxD signals the possible beginning of a START bit and a new character. The START bit is checked by testing for a "low" at its nominal center as specified by the BAUD RATE. If a "low" is detected again, it is considered valid, and the bit assembling counter starts counting. The bit counter locates the approximate center of the date, parity (if specified), and STOP bits. The parity error flag (PE) is set if a parity error occurs. Input bits are sampled at the RxD pin with the rising edge of $\overline{\mathrm{RxC}}$. If a high is not detected for the STOP bit, which normally signals the end of an input character, a framing error (FE) is set. After a valid STOP bit, the input character is loaded into the parallel Data Bus Buffer of the μPD8251 and μPD8251A and the RxRDY signal is raised to indicate to the processor that a character is ready to be fetched. If the processor has failed to fetch the previous character, the new character replaces the old and the overrun flag (OE) is set. All the error flags can be reset by setting a bit in the Command Instruction. Error flag conditions will not stop subsequent USART operation.

The Mode instruction format for Asynchronous Mode is shown in Figure D-7. The Transmit and Receive format for Asynchronous Mode is shown in Figure D-8.



BAUD RATE FACTOR

0	1	0	1
0	0	1	1
SYNC MODE	(1X)	(16X)	(64X)

CHARACTER LENGTH

0	1	0	1
0	0	1	1
5 BITS	6 BITS	7 BITS	8 BITS

PARITY ENABLE
1 = ENABLE 0 = DISABLE

EVEN PARITY GENERATION/CHECK
1 = EVEN 0 = ODD

NUMBER OF STOP BITS

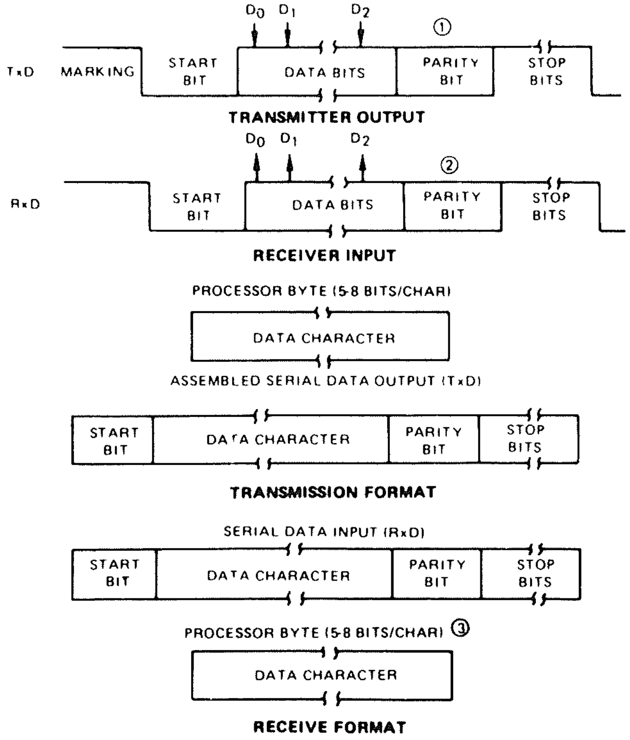
0	1	0	1
0	0	1	1
INVALID	1 BIT	1½ BITS	2 BITS

Figure D-7 Mode Instruction Format Asynchronous Mode

SYNCHRONOUS TRANSMISSION

As in Asynchronous transmission, the TxD output remains "high" (marking) until the μPD8251 and μPD8251A receive the first character (usually a SYNC character) from the processor. After a Command Instruction has set TxEN and after Clear to Send ($\overline{\text{CTS}}$) goes low, the first character is serially transmitted. Data is shifted out on the falling edge of $\overline{\text{TxC}}$ and the same rate as $\overline{\text{TxC}}$.

Once transmission has started, Synchronous Mode format requires that the serial data stream at TxD continue at the $\overline{\text{TxC}}$ rate or SYNC will be lost. If a data character is not provided by the processor before the μPD8251 and μPD8251A



Notes ① Generated by μPD8251/8251A
② Does not appear on the Data Bus.
③ If character length is defined as 5, 6, or 7 bits, the unused bits are set to "zero."

Figure D-8 Transmit/Receive Format Asynchronous Mode

Transmit Buffer becomes empty, the SYNC characters loaded directly following the Mode Instruction are automatically inserted in the TxD data stream. The SYNC characters are inserted to fill the line and maintain synchronization until new data characters are available for transmission. If the μPD8251 and μPD8251A become empty and must send the SYNC characters, the TxEMPTY output is raised to signal the processor that the Transmitter Buffer is empty and SYNC characters are being transmitted. TxEMPTY is automatically reset by the next character from the processor.

SYNCHRONOUS RECEIVE

In Synchronous Receive, character synchronization can be external or internal. If the internal SYNC mode has been selected, and the Enter HUNT (EH) bit has been set by a Command Instruction, the receiver goes into the HUNT mode.

Incoming data on the RxD input is sampled on the rising edge of $\overline{\text{RxC}}$, and the Receive Buffer is compared with the first SYNC character after each bit has been loaded until a match is found. If two SYNC characters have been programmed, the next received character is also compared. When the SYNC characters programmed have been detected, the μPD8251 and μPD8251A leave the HUNT mode and are in character synchronization. At this time, the SYNDET (output) is set high. SYNDET is automatically reset by a STATUS READ.

If external SYNC has been specified in the Mode Instruction, a "one" applied to the SYNDET (input) for at least one $\overline{\text{RxC}}$ cycle synchronizes the USART.

Parity and Overrun Errors are treated the same in the Synchronous as in the Asynchronous Mode. If not in HUNT, parity continues to be checked even if the receiver is not enabled. Framing errors do not apply in the Synchronous format.

The processor may command the receiver to enter the HUNT mode with a Command Instruction that sets Enter HUNT (EH) if synchronization is lost.

The Mode format for Synchronous mode is shown in Figure D-9. The Transmit and Receive format for Synchronous mode is shown in Figure D-10.

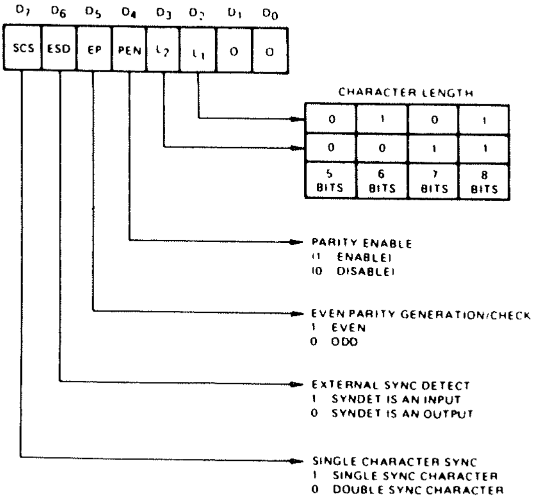
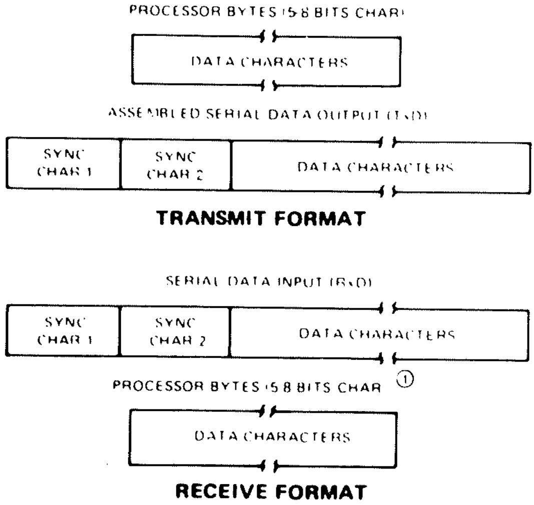


Figure D-9 Mode Instruction Format Synchronous Mode



Note ① If character length is defined as 5, 6 or 7 bits, the unused bits are set to "zero"

Figure D-10 Transmit/Receive Format Synchronous Mode

COMMAND INSTRUCTION FORMAT

After the functional definition of the μPD8251 and μPD8251A has been specified by the Mode Instruction and the SYNC characters have been entered (if in SYNC mode), the USART is ready to receive Command Instructions and begin communication. A Command Instruction controls the specific operation of the format selected by the Mode Instruction. Enable Transmit, Enable Receive, Error Reset and Modem Controls are controlled by the Command Instruction.

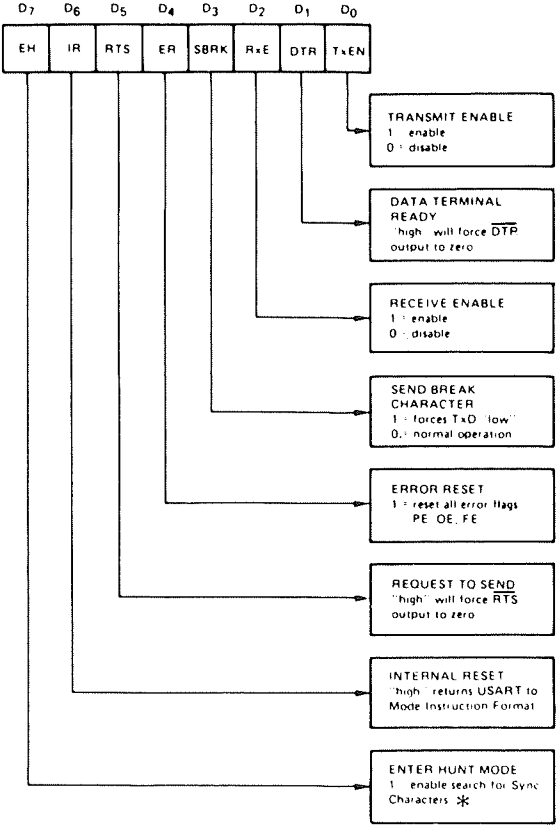
After the Mode Instruction and the SYNC characters (as needed) are loaded, all subsequent "control writes" ($C/\overline{D} = 1$) load or overwrite the Command Instruction register. A Reset operation (internal by CMD IR or external by the RESET input) causes the μPD8251 and μPD8251A to interpret the next "control write", which must immediately follow the reset, as a Mode Instruction.

The Command Instruction format is shown in Figure D-11.

STATUS READ FORMAT

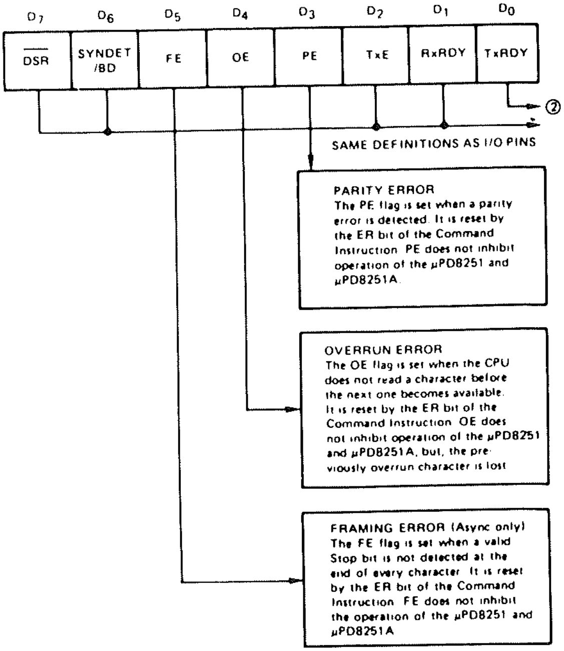
It is frequently necessary for the processor to examine the status of an active interface device to determine if errors have occurred or if there are other conditions that require a response from the processor. The μPD8251 and μPD8251A have features that allow the processor to read the device status at any time. A data fetch is issued by the processor while holding the $C/\overline{D}$ input "high" to obtain device status information. Many of the bits in the status register are copies of external pins. This dual status arrangement allows the μPD8251 and μPD8251A to be used in both polled and interrupt driven environments. Status update can have a maximum delay of 16 clock periods in the μPD8251 and and 28 clock periods in the μPD8251A.

The Status Read format is shown in Figure D-12.



*No effect in ASYNC Mode.

Figure D-11 Command Instruction Format



PARITY ERROR
The PE flag is set when a parity error is detected. It is reset by the ER bit of the Command Instruction. PE does not inhibit operation of the μPD8251 and μPD8251A.

OVERRUN ERROR
The OE flag is set when the CPU does not read a character before the next one becomes available. It is reset by the ER bit of the Command Instruction. OE does not inhibit operation of the μPD8251 and μPD8251A, but, the previously overrun character is lost

FRAMING ERROR (Async only)
The FE flag is set when a valid Stop bit is not detected at the end of every character. It is reset by the ER bit of the Command Instruction. FE does not inhibit the operation of the μPD8251 and μPD8251A

Notes ① No effect in ASYNC mode
② TxRDY status bit is not totally equivalent to the TxRDY output pin, the relationship is as follows
TxRDY status bit = DB Buffer Empty
TxRDY (pin 15) = DB Buffer Empty • $\overline{CTS}$ • TxEn

Figure D-12 Status Read Format

PARITY ERROR

When a parity error is detected, the PE flag is set. It is cleared by setting the ER bit in a subsequent Command Instruction. PE being set does not inhibit USART operation.

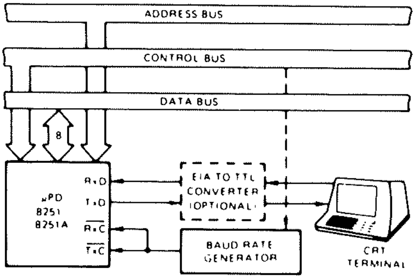
OVERRUN ERROR

If the processor fails to read a data character before the one following is available, the OE flag is set. It is cleared by setting the ER bit in a subsequent Command Instruction. Although OE being set does not inhibit USART operation, the previously received character is overwritten and lost.

FRAMING ERROR*

If a valid STOP bit is not detected at the end of a character, the FE flag is set. It is cleared by setting the ER bit in a subsequent Command Instruction. FE being set does not inhibit USART operation.

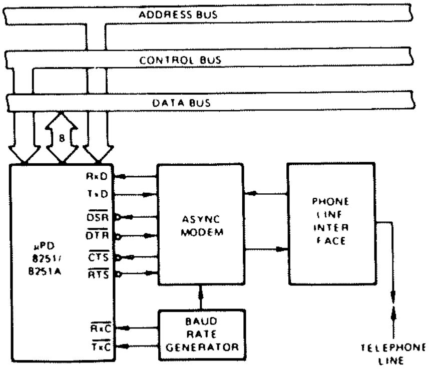
Various applications for the USART are shown in Figure D-13.



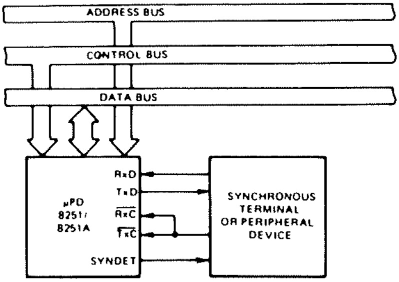
ASYNCHRONOUS SERIAL INTERFACE TO CRT TERMINAL, DC to 9600 BAUD

Figure D-13 Application of the μPD8251 and μPD8251A

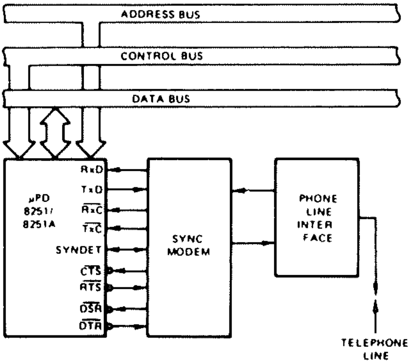
*ASYNC mode only.



ASYNCHRONOUS INTERFACE TO TELEPHONE LINES



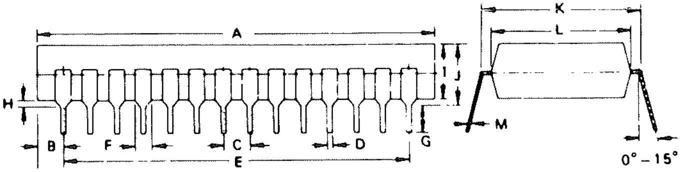
SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE



SYNCHRONOUS INTERFACE TO TELEPHONE LINES

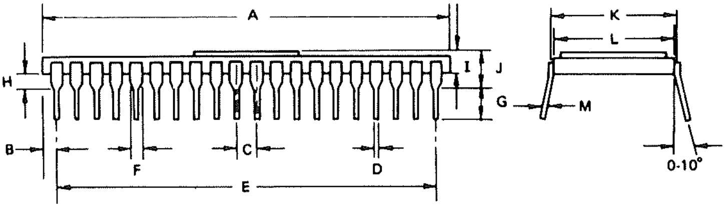
Figure D-13 Application of the μPD8251 and μPD8251A (cont'd)

The outlines for the μPD8251C/D and μPD8251AC/D are shown in Figure D-14.



Plastic

ITEM	MILLIMETERS	INCHES
A	38 0 MAX	1 496 MAX.
B	2 49	0 098
C	2 54	0 10
D	0 5 ± 0 1	0 02 ± 0 004
E	33 02	1 3
F	1 5	0 059
G	2 54 MIN	0 10 MIN
H	0 5 MIN	0 02 MIN
I	5 22 MAX.	0·205 MAX
J	5 72 MAX	0 225 MAX
K	15 24	0 6
L	13 2	0 52
M	0 25 $^{+0.10}_{\ \ 0.05}$	0 01 $^{+0.004}_{\ \ 0.002}$



Ceramic

ITEM	MILLIMETERS	INCHES
A	51.5 MAX.	2.03 MAX.
B	1.62 MAX.	0.06 MAX.
C	2.54 ± 0.1	0.1 ± 0.004
D	0.5 ± 0.1	0.02 ± 0.004
E	48.26 ± 0.1	1.9 ± 0.004
F	1.02 MIN.	0.04 MIN.
G	3.2 MIN.	0.13 MIN.
H	1.0 MIN.	0.04 MIN.
I	3.5 MAX.	0.14 MAX.
J	4.5 MAX.	0.18 MAX.
K	15.24 TYP.	0.6 TYP.
L	14.93 TYP.	0.59 TYP.
M	0.25 ± 0.05	0.01 ± 0.0019

Figure D-14 μD8251C/D and μPD8251AC/D Outlines

APPENDIX E

μPD8255AC-5 PROGRAMMABLE PERIPHERAL INTERFACES

The three-state, bidirectional, 8-bit Data Bus Buffer (D_0-D_7) of the μPD8255 and μPD8255A-5 can be directly interfaced to the processor's system Data Bus (D_0-D_7). The Data Bus Buffer is controlled by execution of IN and OUT instructions by the processor. Control Words and Status information are also transmitted by the Data Bus Buffer.

Read/Write and Control Logic

This block manages all of the internal and external transfers of Data, Control, and Status. Through this block, the processor Address and Control busses can control the peripheral interfaces.

Chip Select, $\overline{CS}$, pin 6

A Logic Low, V_{IL}, on this input enables the μPD8255 and μPD8255A-5 for communication with the 8080A/8085A.

Read, $\overline{RD}$, pin 5

A Logic Low, V_{IL}, on this input enables the μPD8255 and μPD8255A-5 to send Data or Status to the processor via the Data Bus Buffer.

Write, $\overline{WR}$, pin 36

A Logic Low, V_{IL}, on this input enables the Data Bus Buffer to receive Data or Control Words from the processor.

Port Select 0, A_0, pin 9; Port Select 1, A_1, pin 8

These two inputs are used in conjunction with $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$ to control the selection of one of three ports on the Control Word Register. A_0 and A_1 are usually connected to A_0 and A_1 of the processor Address Bus.

Reset, pin 35

A Logic High, V_{IH}, on this input clears the Control Register and sets ports A, B, and C to the input mode. The input latches in ports A, B, and C are not cleared.

Group I and Group II Controls

Through an OUT instruction in system software from the processor, a control word is transmitted to the µPD8255 and µPD8255A-5. Information such as "MODE," "Bit SET," and "Bit RESET" is used to initialize the functional configuration of each I/O port.

Each group (I and II) accepts "commands" from the Read/Write Control Logic and "control words" from the internal data bus and in turn controls its associated I/O ports.

Group I — Port A and upper Port C (PC_7-PC_4)

Group II — Port B and lower Port C (PC_3-PC_0)

While the Control Word Register can be written into, the contents cannot be read back to the processor.

Ports A, B, and C

The three 8-bit I/O ports (A, B, and C) in the µPD8255 and µPD8255A-5 can all be configured to meet a wide variety of functional requirements through system software. The effectiveness and flexibility of the µPD8255 and µPD8255A-5 is further enhanced by special features unique to each of the ports.

Port A = An 8-bit data output latch/buffer and data input latch.

Port B = An 8-bit data input/output latch/buffer and an 8-bit data input buffer.

FUNCTIONAL DESCRIPTION

General

The μPD8255 and μPD8255A-5 Programmable Peripheral Interfaces (PPI) are designed for use in 8080A/8085A microprocessor systems. Peripheral equipment can be effectively and efficiently interfaced to the 8080A/8085A data and control busses with the μPD8255 and μPD8255A-5. The μPD8255 and μPD8255A-5 are functionally configured to be programmed by system software to avoid external logic for peripheral interfaces.

Data Bus Buffer

The three-state, bidirectional, 8-bit Data Bus Buffer (D_0-D_7) of the μPD8255 and μPD8255A-5 can be directly interfaced to the processor's system Data Bus (D_0-D_7). The Data Bus Buffer is controlled by execution of IN and OUT instructions by the processor. Control Words and Status information are also transmitted by the Data Bus Buffer.

Read/Write and Control Logic

This block manages all of the internal and external transfers of Data, Control, and Status. Through this block, the processor Address and Control busses can control the peripheral interfaces.

Chip Select, $\overline{\text{CS}}$, pin 6

A Logic Low, V_{IL}, on this input enables the μPD8255 and μPD8255A-5 for communication with the 8080A/8085A.

Read, $\overline{\text{RD}}$, pin 5

A Logic Low, V_{IL}, on this input enables the μPD8255 and μPD8255A-5 to send Data or Status to the processor by the Data Bus Buffer.

Write, $\overline{\text{WR}}$, pin 36

A Logic Low, V_{IL}, on this input enables the Data Bus Buffer to receive Data or Control Words from the processor.

Port Select 0, A_0, pin 9; Port Select 1, A_1, pin 8

These two inputs are used in conjunction with $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$ to control the selection of one of three ports on the Control WordRegister. A_0 and A_1 are usually connected to A_0 and A_1 of the processor Address Bus.

Reset, pin 35

A Logic High, V_{IH}, on this input clears the Control Register and sets ports A, B, and C to the input mode. The input latches in ports A, B, and C are not cleared.

Group I and Group II Controls

Through an OUT instruction in system software from the processor, a control word is transmitted to the µPD8255 and µPD8255A-5. Information such as "MODE," "Bit SET," and "Bit RESET" is used to initialize the functional configuration of each I/O port.

Each group (I and II) acepts "commands" from the Read/Write Control Logic and "control words" from the internal data bus and in turn controls its associated I/O ports.

Group I — Port A and upper Port C (PC_7-PC_4)

Group II — Port B and lower Port C (PC_3-PC_0)

While the Control Word Register can be written into, the contents cannot be read back to the processor.

Ports A, B, C

The three 8-bit I/O ports (A, B, and C) in the µPD8255 and µPD8255A-5 can all be configured to meet a wide variety of functional requirements through system software. The effectiveness and flexibility of the µPD8255 and µPD825A-5 are further enhanced by special features unique to each of the ports.

Port A = An 8-bit data output latch/buffer and data input latch.

Port B = An 8-bit data input/output latch/buffer and an 8-bit data input buffer.

Port C = An 8-bit output latch/buffer and a data input buffer (input not latched).

Port C can be divided into two independent 4-bit control and status ports for use with Ports A and B.

A block diagram of Ports A, B and C is shown in Figure E-1.

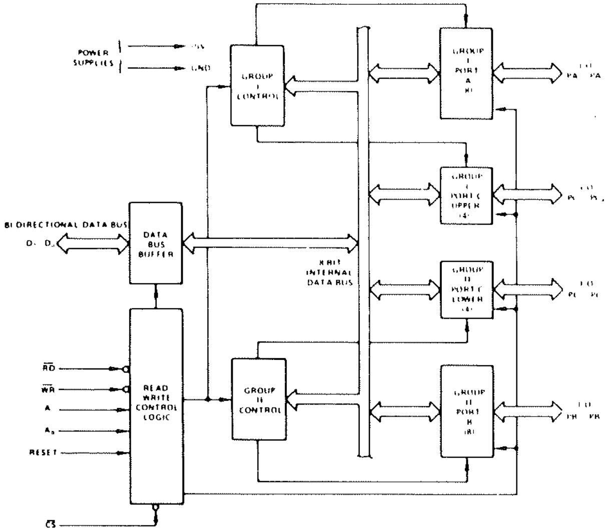


Figure E-1   Block Diagram

ABSOLUTE MAXIMUM RATINGS

Operating Temperature	0°C to +70°C
Storage Temperature	–65°C to +125°C
All Output Voltages*	–0.5 to +7 Volts
All Input Voltages*	–0.5 to +7 Volts
Supply Voltages*	–0.5 to +7 Volts

*With respect to V_{SS}

COMMENT: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*T_a = 25°C

The DC characteristics, capacitance and AC characteristics for the µPD8255 are given in Tables E-1 through E-3.

Table E-1 DC Characteristics

T_a = 0°C to +70°C, V_{CC} = +5V ± 10%, V_{SS} = 0V

PARAMETER	SYMBOL	LIMITS						UNIT	TEST CONDITIONS
		µPD8256			µPD8255A 5				
		MIN	TYP	MAX	MIN	TYP	MAX		
Input Low Voltage	V_{IL}	V_{SS}-0 5		0 8	-0 5		0 8	V	
Input High Voltage	V_{IH}	2		V_{CC}	2		V_{CC}	V	
Output Low Voltage	V_{OL}			0 4			0 45	V	②
Output High Voltage	V_{OH}	2 4			2 4			V	③
Darlington Drive Current	I_{OH} ①	1	2	4	-1		-4	mA	V_{OH} 1 5V, R_{EXT} 750Ω
Power Supply Current	I_{CC}		40	120			120	mA	V_{CC} = +5V Output Open
Input Leakage Current	I_{LIH}			10			10	µA	V_{IN} - V_{CC}
Input Leakage Current	I_{LIL}			-10			-10	µA	V_{IN} 0 4V
Output Leakage Current	I_{LOH}			10			±10	µA	V_{OUT} - V_{CC}, $\overline{CS}$ = 2 0V
Output Leakage Current	I_{LOL}			-10			-10	µA	V_{OUT} = 0 4V, $\overline{CS}$ 2 0V

Notes ① Any set of eight (8) outputs from either Port A, B or C can source 2 mA into 1 5V for µPD8255, or 4 mA into 1 5V for µPD8255A 5

② For µPD8255 I_{OL} 1 7 mA
For µPD8255A 5 I_{OL} 2 5 mA for DB Port 1 7 mA for Peripheral Ports

③ For µPD8255 I_{OH} = - 100 µA for DB Port; 50 µs for Peripheral Ports.
For µPD8255A-5 I_{OH} = -400 µA for dB Port; - 200 µs for Peripheral Ports.

Table E-2 Capacitance

T_a = 25°C, V_{CC} = V_{SS} = 0V

PARAMETER	SYMBOL	LIMITS			UNIT	TEST CONDITIONS
		MIN	TYP	MAX		
Input Capacitance	C_{IN}			10	pF	f_c = 1 MHz
I/O Capacitance	$C_{I/O}$			20	pF	Unmeasured pins returned to V_{SS}

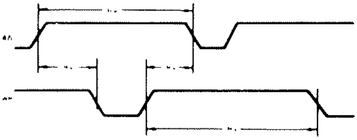
Table E-3 AC Characteristics

$T_a = 0°C$ to $+70°C$ $V_{CC} = +5V \pm 5\%$ $V_{SS} = 0V$

PARAMETER	SYMBOL	LIMITS				UNIT	TEST CONDITIONS
		μPD8255		μPD8255A-5			
		MIN	MAX	MIN	MAX		
READ							
Address Stable Before $\overline{\text{READ}}$	t_{AR}	50		0		ns	
Address Stable After $\overline{\text{READ}}$	t_{RA}	0		0		ns	
$\overline{\text{READ}}$ Pulse Width	t_{RR}	405		300		ns	
Data Valid From $\overline{\text{READ}}$	t_{RD}		295		200	ns	8255 $C_L = 100$ pF 8256A-5 $C_L = 150$ pF
Data Float After $\overline{\text{READ}}$	t_{DF}	 10	150	 10	100	ns ns	$C_L = 100$ pF $C_L = 15$ pF
Time Between $\overline{\text{READS}}$ and/or $\overline{\text{WRITES}}$	t_{RV}	850		850		ns	②
WRITE							
Address Stable Before $\overline{\text{WRITE}}$	t_{AW}	20		0		ns	
Address Stable After $\overline{\text{WRITE}}$	t_{WA}	20		20		ns	
$\overline{\text{WRITE}}$ Pulse Width	t_{WW}	400		300		ns	
Data Valid To $\overline{\text{WRITE}}$ (L.E.)	t_{DW}	10		100		ns	
Data Valid After $\overline{\text{WRITE}}$	t_{WD}	35		3u		ns	
OTHER TIMING							
$\overline{\text{WR}}$ = 0 To Output	t_{WB}		500		350	ns	8255 $C_L = 50$ pF 8256A 5 $C_L = 150$ pF
Peripheral Data Before $\overline{\text{RD}}$	t_{IR}	0		0		ns	
Peripheral Data After $\overline{\text{RD}}$	t_{HR}	50		0		ns	
$\overline{\text{ACK}}$ Pulse Width	t_{AK}	500		300		ns	
$\overline{\text{STB}}$ Pulse Width	t_{ST}	350		500		ns	
Per. Data Before T.E. Of $\overline{\text{STB}}$	t_{PS}	60		0		ns	
Per. Data After T.E. Of $\overline{\text{STB}}$	t_{PH}	150		180		ns	
$\overline{\text{ACK}}$ = 0 To Output	t_{AD}		400		300	ns	8255 $C_L = 50$ pF 8255A-5 $C_L = 150$ pF
$\overline{\text{ACK}}$ = 0 To Output Float	t_{KD}	 20	300	 20	250	ns	8255 $\begin{cases} C_L = 50\text{ pF} \\ C_L = 15\text{ pF} \end{cases}$
$\overline{\text{WR}}$ = 1 To $\overline{\text{OBF}}$ = 0	t_{WOB}		300		650	ns	8256 $C_L = 50$ pF 8255A 5 $C_L = 150$ pF
$\overline{\text{ACK}}$ = 0 To $\overline{\text{OBF}}$ = 1	t_{AOB}		450		350	ns	
$\overline{\text{STB}}$ = 0 To IBF = 1	t_{SIB}		450		300	ns	
$\overline{\text{RD}}$ = 1 To IBF = 0	t_{RIB}		360		300	ns	
$\overline{\text{RD}}$ = 0 To INTR = 0	t_{RIT}		450		400	ns	
$\overline{\text{STB}}$ = 1 To INTR = 1	t_{SIT}		400		300	ns	
$\overline{\text{ACK}}$ = 1 To INTR = 1	t_{AIT}		400		350	ns	
$\overline{\text{WR}}$ = 0 To INTR = 0	t_{WIT}		850		850	ns	

Notes: ① Period of Reset pulse must be at least 50 μs during or after power on. Subsequent Reset pulse can be 500 ns min.

②



Various timing waveforms for the μPD8255 are given in Figure E-2.

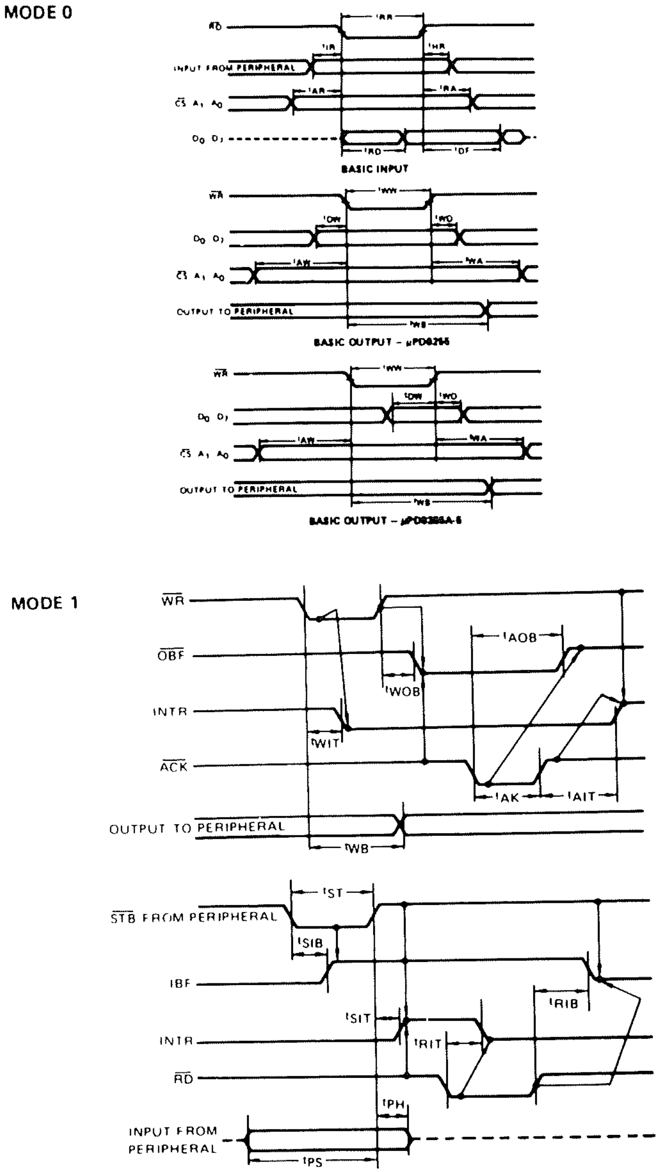
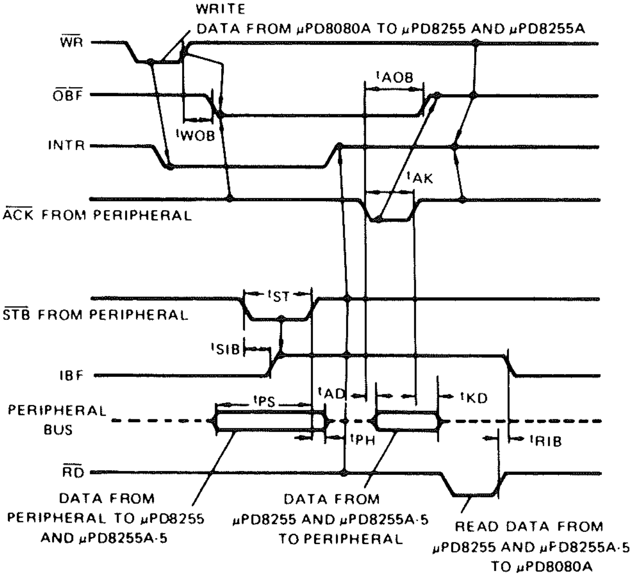


Figure E-2 Timing Waveforms

MODE 2


WRITE
DATA FROM μPD8080A TO μPD8255 AND μPD8255A
$\overline{WR}$
$\overline{OBF}$
INTR
$\overline{ACK}$ FROM PERIPHERAL
t_{AOB}
t_{WOB}
t_{AK}
$\overline{STB}$ FROM PERIPHERAL
t_{ST}
t_{SIB}
IBF
PERIPHERAL BUS
t_{PS}
t_{AD}
t_{KD}
t_{PH}
t_{RIB}
$\overline{RD}$
DATA FROM PERIPHERAL TO μPD8255 AND μPD8255A-5
DATA FROM μPD8255 AND μPD8255A-5 TO PERIPHERAL
READ DATA FROM μPD8255 AND μPD8255A-5 TO μPD8080A

Note: ① Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF · $\overline{MASK}$ · $\overline{STB}$ · $\overline{RD}$ + $\overline{OBF}$ · $\overline{MASK}$ · $\overline{ACK}$ · $\overline{WR}$)

② When the μPD8255A-5 is set to Mode 1 or 2, $\overline{OBF}$ is reset to be high (logic 1).

Figure E-2 Timing Waveforms (cont'd)

MODES

The μPD8255 and μPD8255A-5 can be operated in modes (0, 1 or 2), which are selected by appropriate control words and are detailed below.

MODE 0

MODE 0 provides for basic Input and Output operations through each of the ports A, B, and C. Output data is latched and input data follows the peripheral. No "handshaking" strobes are needed.

16 different configurations in MODE 0.

Two 8-bit ports and two 4-bit ports.

Inputs are not latched.

Outputs are latched.

MODE 1

MODE 1 provides for Strobed Input and Output operations with data transferred through Port A or B and handshaking through Port C.

Two I/O Groups (I and II).

Both groups contain an 8-bit data port and a 4-bit control/data port.

Both 8-bit data ports can be either Latched Input or Latched Output.

MODE 2

MODE 2 provides for Strobed bidirectional operation using PA_{0-7} as the bidirectional latched data bus. PC_{3-7} is used for interrupts and "handshaking" bus flow controls similar to Mode 1. Note that PB_{0-7} and PC_{0-2} may be defined as Mode 0 or 1, input or output in conjunction with Port A in Mode 2.

An 8-bit latched bidirectional bus port (PA_{0-7}) and a 5-bit control port (PC_{3-7})

Both inputs and outputs are latched.

An additional 8-bit input or output port with a 3-bit control port.

The basic operation of the μPD8255 is shown in Table E-3.

Table E-3 Basic Operation

INPUT OPERATION (READ)					
A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	0	1	0	PORT A → DATA BUS
0	1	0	1	0	PORT B → DATA BUS
1	0	0	1	0	PORT C → DATA BUS

OUTPUT OPERATION (WRITE)					
A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	0	DATA BUS → PORT A
0	1	1	0	0	DATA BUS → PORT B
1	0	1	0	0	DATA BUS → PORT C
1	1	1	0	0	DATA BUS → CONTROL

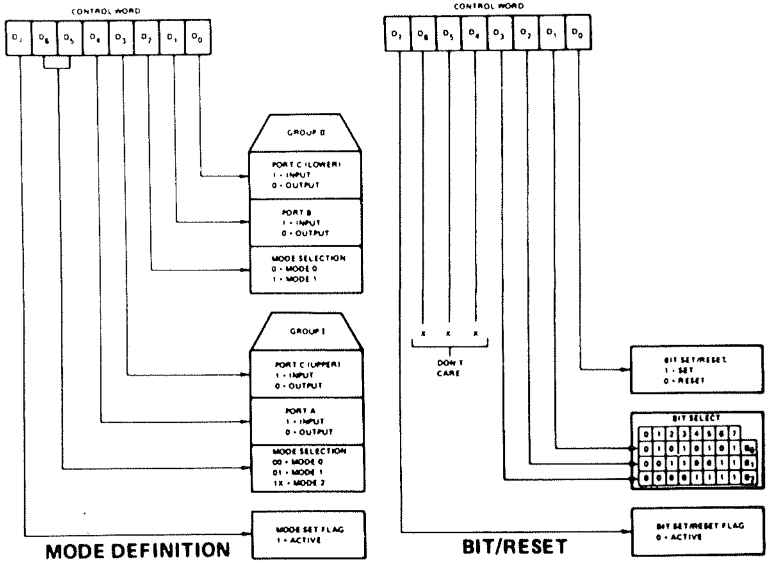
DISABLE FUNCTION					
A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
X	X	X	X	1	DATA BUS → HIGH Z STATE
X	X	1	1	0	DATA BUS → HIGH Z STATE

NOTES ① X means "DO NOT CARE"

② All conditions not listed are illegal and should be avoided

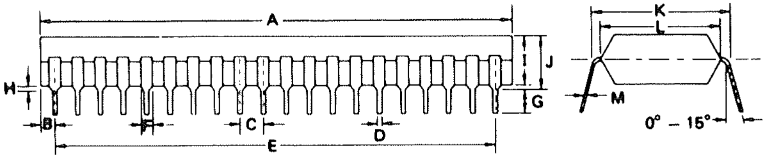
The formats of the μPD8255 are diagrammed in Figure E-3.

Figure E-3 Formats



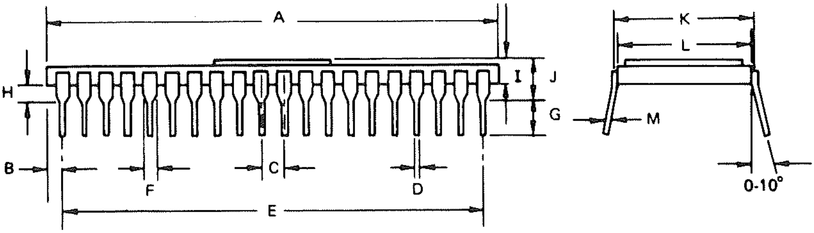
0	1	2	3	4	5	6	7	
0	1	0	1	0	1	0	1	B_0
0	0	1	1	0	0	1	1	B_1
0	0	0	0	1	1	1	1	B_2

The outline for the μPD8255 is shown in Figure E-4.



Plastic

ITEM	MILLIMETERS	INCHES
A	51.5 MAX	2.028 MAX
B	1.62	0.064
C	2.54 ± 0.1	0.10 ± 0.004
D	0.5 ± 0.1	0.019 ± 0.004
E	48.26	1.9
F	1.2 MIN	0.047 MIN
G	2.54 MIN	0.10 MIN
H	0.5 MIN	0.019 MIN
I	5.22 MAX	0.206 MAX
J	5.72 MAX	0.225 MAX
K	15.24	0.600
L	13.2	0.520
M	0.25 $^{+0.1}_{-0.05}$	0.010 $^{+0.004}_{0.002}$



Ceramic

ITEM	MILLIMETERS	INCHES
A	51.5 MAX.	2.03 MAX.
B	1.62 MAX.	0.06 MAX.
C	2.54 ± 0.1	0.1 ± 0.004
D	0.5 ± 0.1	0.02 ± 0.004
E	48.26 ± 0.1	1.9 ± 0.004
F	1.02 MIN.	0.04 MIN.
G	3.2 MIN.	0.13 MIN.
H	1.0 MIN.	0.04 MIN.
I	3.5 MAX.	0.14 MAX.
J	4.5 MAX.	0.18 MAX.
K	15.24 TYP.	0.6 TYP.
L	14.93 TYP.	0.59 TYP.
M	0.25 ± 0.05	0.01 ± 0.0019

Figure E-4 μPD8255C and μPD8255AC/D-5 Outline

NEC

Printed in Japan
78110371